



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y
DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica y Automática Industrial

TRABAJO FIN DE GRADO

DISEÑO Y CONSTRUCCIÓN DE UN HELICÓPTERO COAXIAL

Eduardo Ortega Biber

Cotutor: D. Alberto Brunete
González

Departamento: Electrónica
Automática e Informática
Industrial

Tutor: D. Miguel Hernando
Gutiérrez

Departamento: Electrónica
Automática e Informática
Industrial

Madrid, Julio 2015



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y
DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica y Automática Industrial

TRABAJO FIN DE GRADO

DISEÑO Y CONSTRUCCIÓN DE UN
HELICÓPTERO COAXIAL

Firma Autor

Firma Cotutor

Firma Tutor

Título: Diseño y construcción de un helicóptero coaxial

Autor: Eduardo Ortega Biber

Tutor: D. Miguel Hernando Gutiérrez

Cotutor: D. Alberto Brunete González

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día de de ... en, en la Escuela Técnica Superior de Ingeniería y Diseño Industrial de la Universidad Politécnica de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Agradezco a mis padres y mis hermanos el apoyo recibido.

A Nicolás Parra, mi compañero de proyecto, por acompañarme durante la realización de este trabajo.

A Miguel Hernando, mi tutor, su dedicación prestada, la orientación recibida y por transmitirme el entusiasmo por la robótica.

Agradezco a Alberto Brunete, cotutor del trabajo, su ayuda prestada.

Resumen

Este proyecto sienta las bases para el desarrollo de un helicóptero coaxial autónomo. Este helicóptero consta de dos hélices montadas sobre un mismo eje y con sentidos de rotación opuestos. Para manejar el helicóptero, este trabajo propone un mecanismo capaz de desplazar el centro de gravedad del helicóptero. El control se realizará mediante los sensores de un teléfono móvil montado en el helicóptero. Este teléfono proporcionará además diferentes recursos para poder desarrollar futuras aplicaciones. Una aplicación ejecutada en un segundo teléfono móvil permitirá enviar las órdenes para maniobrar el helicóptero.

El proyecto está dividido en dos mitades presentándose en el presente Trabajo Fin de Grado el diseño y la construcción del helicóptero.

La segunda parte de este proyecto, referente al desarrollo de las aplicaciones para los dos teléfonos móviles, es abordada por Nicolás Parra Sánchez.

Palabras clave: helicóptero, coaxial, autónomo.

Abstract

This thesis lays the foundations for the development of an autonomous coaxial helicopter. This is a helicopter with two propellers mounted on the same axis and revolving in opposites directions. To steer the helicopter, this thesis proposes a mechanism that moves the helicopter's centre of mass. The control will be made by the sensors of a mobile phone mounted on the helicopter. This phone will also provide various resources for the development of future applications. An application running on a second mobile phone will let you send commands to maneuver the helicopter.

The thesis is divided into two halves, presenting in this paper the design and construction of the helicopter.

The second part of this project, concerning the development of the applications for the two mobile phones, is approached by Nicolás Parra Sánchez.

Keywords: autonomous, coaxial, helicopter.

Índice general

Agradecimientos	v
Resumen	vii
Abstract	ix
1. Introducción	1
1.1. Motivación del proyecto	1
1.2. Objetivos	2
1.3. Materiales utilizados	2
1.4. Estructura del documento	3
2. Estado del arte	5
2.1. Vehículos aéreos no tripulados	5
2.2. Helicóptero coaxial	6
2.3. Configuración, manejo y control	7
2.4. Proyectos relacionados	8
3. Componentes del helicóptero	13
3.1. Motores	13
3.2. Batería	14
3.3. Controladores electrónicos	15
3.4. Hélices	15
3.5. Servomotores	16
3.6. Elección de componentes	17
3.7. Arduino®	18
3.8. Zum BT-328	20
4. Pruebas y prototipos	23
4.1. Fresado CNC	23
4.2. Impresión 3D	24
4.3. Banco de pruebas	26
4.4. Conexión de los componentes	27
4.5. Programa Arduino	28
4.6. Placa PCB	29
4.7. Placa BQ	30
4.8. Prototipos del helicóptero	31

5. Registro de movimiento	35
5.1. Láser Hokuyo	35
5.2. Protocolo SCIP2.0	36
5.3. Captura de datos	38
5.3.1. UrgBenri y Excel	38
5.3.2. MATLAB	39
6. Simulación del modelo dinámico del helicóptero	43
6.1. Simulink	43
6.1.1. SimMechanics	43
6.2. Regulador PID	47
6.3. Cálculo del empuje de los rotores	49
6.3.1. Teoría de la cantidad de movimiento [21]	49
6.3.2. Análisis dimensional [26]	50
7. Aplicación BQ ZUM	55
7.1. Protocolo de comunicación	55
7.2. Sensor ultrasónico	57
7.3. Control de velocidad	60
7.4. Bucle de la aplicación	62
8. Diseño del helicóptero	63
8.1. Diseño	63
8.2. Control redundante de seguridad	65
9. Gestión del proyecto	71
9.1. Ciclo de vida	71
9.2. Planificación	72
9.3. Presupuesto	72
9.3.1. Personal	72
9.3.2. Material	72
9.3.3. Resumen de costes	72
10. Conclusiones	75
10.1. Conclusión	75
10.2. Desarrollos futuros	76
A. Anexos	77
Bibliografía	101

Índice de figuras

2.1. Helicópteros no tripulados	6
2.2. Comparación helicóptero convencional con helicóptero coaxial	7
2.3. Montaje de las aspas con bisagras	8
2.4. Plato oscilante del helicóptero Lama	8
2.5. Mecanismo de desplazamiento del centro de gravedad	9
2.6. Helicóptero micro UAV	9
2.7. Prototipo helicóptero coaxial con desplazamiento del centro de gravedad	10
2.8. Helicóptero Gimball de Flyability	10
2.9. Plataforma Flone para teléfonos móviles	11
3.1. Motor sin escobillas CR28M	14
3.2. Batería LiPo Rhino 4900mAh	14
3.3. Densidad de carga de los diferentes tipos de baterías	15
3.4. Controlador electrónico Turnigy Plush 25A	16
3.5. Servomotor HD-3001HB	16
3.6. Oferta motores coaxiales	17
3.7. Oferta baterías 3S1P	18
3.8. Oferta variadores electrónicos de 25 A	19
3.9. Arduino Yún y librería Bridge	20
3.10. Placa ZUM BT-328	21
4.1. Diseño piezas en AutoCAD	24
4.2. Impresora 3D Prusa i3	25
4.3. Diseño carcasa para la batería	25
4.4. Diseño del primer banco de pruebas	26
4.5. Primer banco de pruebas	27
4.6. Segundo banco de pruebas	27
4.7. Conexión de conector tipo servo	28
4.8. Interfaz gráfica en HTML	29
4.9. Placa PCB	30
4.10. Interfaz de la aplicación ArduinoRC	31
4.11. Brazo protección ABS	32
4.12. Prototipo protegido con varillas de carbono	32
4.13. Prototipos helicóptero	34
4.14. Prototipo híbrido con protección	34
5.1. Láser Hokuyo UTM-30LX	35
5.2. Área detección UTM-30LX	36
5.3. Codificación de distancias en protocolo SCIP2.0	37

5.4.	Descodificación de distancias en protocolo SCIP2.0	37
5.5.	Comunicación entre servidor y sensor	37
5.6.	Comando ME	38
5.7.	Programa URGBenri	38
5.8.	Medidas mostradas en aplicación Excel	39
5.9.	Ensayo con el helicóptero	41
5.10.	Nube de puntos en la aplicación MATLAB	41
5.11.	Nube de puntos en la aplicación MATLAB	42
6.1.	Visualización de la simulación en SimMechanics Explorer y salidas del sistema	44
6.2.	Modelo empuje hélices	45
6.3.	Actuadores en el modelo	46
6.4.	Sistema de helicóptero y control en Simulink	47
6.5.	Regulador PID	48
6.6.	Aplicación PID Tuner y Bloques PIDs modelo	49
6.7.	Parámetros hélices APC 10x6	53
7.1.	Composición de un mensaje del protocolo	56
7.2.	Diagrama del protocolo de comunicación	57
7.3.	Sensor de ultrasonidos BAT	58
7.4.	Secuencia de funcionamiento del sensor BAT	58
7.5.	Diagrama de la medición con el sensor de ultrasonidos	59
7.6.	Medidas a medio metro de altura con el sensor BAT	60
7.7.	Medidas a un metro de altura con el sensor BAT	60
7.8.	Medidas de velocidad con el sensor BAT	60
7.9.	Diagrama del control de velocidad	61
7.10.	Salidas del controlador P	62
8.1.	Diseño final helicóptero coaxial	64
8.2.	Patín del tren de aterrizaje	64
8.3.	Detalle de mecanismo para desplazar el centro de gravedad	65
8.4.	Detalle de la sujeción del teléfono móvil	65
8.5.	Detalle del mecanismo para desplazar la batería	66
8.6.	Jaula para ensayos de vuelo	66
8.7.	Diseño esquemático de la placa de conmutación	68
8.8.	Diseño físico de la placa de conmutación	69
9.1.	Evolución temporal del proyecto	73

Índice de tablas

3.1.	Comparación entre diferentes tipos de baterías	15
3.2.	Pesos medios de los componentes	17
3.3.	Especificaciones motor CR28M	18
3.4.	Especificaciones batería Rhino 4900mAh 3S1P	18
3.5.	Especificaciones controladores electrónicos Turnigy Plush 25 A	19
3.6.	Especificaciones servomotor HD - 3001HB	19
3.7.	Especificaciones Arduino Yún	20
3.8.	Especificaciones BQ ZUM BT-328	21
5.1.	Especificaciones láser Hokuyo UTM-30LX	36
6.1.	Parámetros análisis dimensional	51
7.1.	Especificaciones sensor ultrasonidos BAT	58
8.1.	Tabla de verdad del multiplexor 74LS157	67
9.1.	Presupuesto material del proyecto	73

Capítulo 1

Introducción

1.1. Motivación del proyecto

El motivo principal de la elección del presente Trabajo Fin de Grado nace del interés por realizar un proyecto multidisciplinar en el que se pudieran poner en práctica el mayor número de aptitudes y conocimientos adquiridos durante el estudio del Grado en Ingeniería Electrónica Industrial y Automática. También debía suponer en cierta medida un reto que incitara a continuar la formación permitiendo el aprendizaje de tecnologías aún sin conocer.

El auge actual de los vehículos aéreos no tripulados provoca también la predilección por esta rama de la robótica. El descenso en el precio de los sensores y la llegada de rápidos microcontroladores a precios reducidos ha permitido que esta especialidad abandone los centros de investigación y pueda ser estudiada sin necesitar grandes inversiones. Además cuenta con la integración de las principales materias estudiadas como la informática, la regulación automática y la electrónica.

Por otro lado, la idea de innovar con el trabajo y buscar un nuevo enfoque a los populares cuadricópteros origina la idea del helicóptero coaxial. Al no estar tan divulgado su desarrollo supone un desafío aún mayor.

También se tiene en mente la idea de construir un producto nuevo y universal que sirva como plataforma para el desarrollo de un gran número de futuras aplicaciones. Esto se permite mediante la incorporación de un teléfono móvil en la aeronave. Una de las ventajas de esta propiedad reside en el ahorro de sensores adicionales para el control de vuelo debido a que están incorporados en los actuales teléfonos móviles. Pero esta particularidad no es la principal. La multitud de características y tecnologías presentes en estos aparatos, desde la cámara de vídeo y fotos a los diferentes sensores como el GPS, habilitan la posibilidad de desarrollar abundantes aplicaciones para el helicóptero.

Además la rápida y constante evolución de estos dispositivos móviles permitirá, en poco tiempo, añadir nuevas funcionalidades que aún no se han imaginado. Este avance tecnológico se percibe en el interés de empresas como Google, Apple y Qualcomm en diseñar teléfonos con sensores como cámaras 3-D y unidades de mediciones inerciales (IMU). Un ejemplo de estos terminales es el proyecto Tango de la empresa Google [27]. El grupo de Proyectos y Tecnologías Avanzadas (ATAP, por sus siglas en inglés) de esta empresa está desarrollando un teléfono móvil inteligente capaz de crear un modelo tridimensional del espacio que le rodea. Para ello utiliza una tecnología similar al sensor Kinect[®] [19] desarrollado por Microsoft. Actualmente el

teléfono Tango se encuentra en la etapa de desarrollo con lo que aún no se conoce el alcance total de su funcionalidad. Pero la intención del grupo de Proyectos y Tecnologías Avanzadas es crear aplicaciones para navegación interior y exterior, lo que aumentaría enormemente las capacidades del helicóptero coaxial. Esta característica ha sido puesta en práctica en [15] al incorporar un teléfono del proyecto Tango en un cuadricóptero y conseguir realizar vuelos autónomos dentro de un laboratorio.

1.2. Objetivos

Este trabajo pretende sentar las bases para el desarrollo de un helicóptero coaxial. El helicóptero debe servir como plataforma para la creación de futuras aplicaciones.

Como objetivos principales están el diseño y la construcción de un helicóptero coaxial controlado por un mecanismo que desplace el centro de gravedad.

Para alcanzarlo se establece la siguiente progresión de objetivos:

- Selección de los componentes físicos necesarios para la construcción de un helicóptero coaxial.
- Diseño y construcción de un banco de pruebas que sirva como base para realizar ensayos con los componentes.
- Desarrollo de una aplicación para registrar el movimiento de la aeronave con el fin de conocer su respuesta ante diferentes entradas de control.
- Creación de un modelo dinámico computarizado del helicóptero para su simulación.
- Diseño y construcción de un helicóptero coaxial.
- Programación de una aplicación que se comunique con un teléfono móvil y controle el helicóptero.

1.3. Materiales utilizados

Durante la elaboración de este trabajo se han empleado diferentes instrumentos proporcionados por el Departamento de Electrónica, Automática e Informática Industrial. Entre ellos se encuentran las herramientas utilizadas para la fabricación del helicóptero, como la impresora Hephestos i3 y la máquina fresadora CNC. La impresora se emplea para producir las piezas en plástico ABS, mientras que las piezas de metacrilato se realizan con la fresadora.

Para construir el helicóptero se aprovechan las herramientas, llaves y demás utensilios, del laboratorio de robótica perteneciente al departamento. El láser Hokuyo UTM-30LX empleado en la aplicación para registrar el movimiento del helicóptero se encuentra asimismo en este laboratorio.

También se aprovechan una fuente de tensión para alimentar los circuitos diseñados, un soldador para realizar las uniones de los componentes de estos circuitos y un multímetro para realizar las medidas convenientes.

Como herramientas principales empleadas en este proyecto se encuentran un ordenador personal con sistema operativo Windows 7 Professional N y un ordenador portátil con una distribución de OS X Yosemite. El primero se utiliza en las tareas

de creación de diseños con herramientas de diseño asistido por computadora, como AutoCAD y Autodesk Inventor Professional, y en el desarrollo de aplicaciones y simulaciones con los programas MATLAB y Simulink. El ordenador portátil se destina a la programación de aplicaciones en el entorno de desarrollo Arduino IDE.

1.4. Estructura del documento

A continuación y para facilitar la lectura del documento, se detalla el contenido de cada capítulo.

- En el primer capítulo se introduce al lector en el proyecto indicando los motivos de su elaboración así como su alcance.
- En el segundo capítulo se explica la evolución de los vehículos aéreos no tripulados describiéndose las características de su variante coaxial. Se exponen asimismo proyectos relacionados con el presente trabajo.
- El tercer capítulo recoge los diferentes componentes necesarios para efectuar el montaje de un helicóptero autónomo incluyendo una breve descripción de cada uno de ellos.
- En el cuarto apartado se relata la construcción del banco de pruebas y los primeros prototipos. Se explican también en él las diferentes tecnologías utilizadas durante el proceso de fabricación.
- En el capítulo número cinco se detalla la aplicación para registrar el movimiento de la aeronave. Para ello se presenta el telémetro láser utilizado y se reseña su protocolo de comunicación. Igualmente se recoge el desarrollo de la aplicación creada para realizar las medidas.
- A continuación, en el capítulo sexto, se expone el software utilizado para la simulación por ordenador del helicóptero. Se aclara aquí la finalidad de esta simulación así como el controlador utilizado para el vuelo de la aeronave.
- El séptimo apartado describe la aplicación creada para ser ejecutada en el helicóptero con el fin de poder manejar y controlar el vehículo.
- En el octavo capítulo se muestra el diseño final del helicóptero.
- En el noveno capítulo se detalla el ciclo de vida del proyecto y se desglosa su presupuesto.
- El capítulo décimo sirve para mostrar las conclusiones y presentar posibles desarrollos futuros del proyecto.

Capítulo 2

Estado del arte

En este capítulo se introducen los vehículos aéreos no tripulados y se describe el helicóptero coaxial junto con sus ventajas frente a diferentes configuraciones de helicópteros.

2.1. Vehículos aéreos no tripulados

Los vehículos aéreos no tripulados (VANT, o UAV por sus siglas en inglés - Unmanned Aerial Vehicle) son aeronaves que vuelan sin tripulación humana a bordo. El avión no tripulado más antiguo fue desarrollado después de la Primera Guerra Mundial y se emplearon durante la Segunda Guerra Mundial para entrenar a los operarios de los cañones antiaéreos [5].

Históricamente los vehículos aéreos no tripulados eran principalmente aviones pilotados de forma remota y usados con fines militares, tanto de reconocimiento como de ataque, donde son denominados vehículos aéreos de combate no tripulados (UCAV por su nombre en inglés).

Con el avance de la tecnología de los microprocesadores y la mejora de las capacidades de las baterías, ha aumentado el desarrollo de este tipo de vehículos. En la actualidad se utilizan cada vez más en aplicaciones civiles de vigilancia, seguridad o incluso educativos y recreativos.

Los vehículos aéreos se pueden clasificar en dos grandes divisiones: de ala fija o de ala giratorias. Dentro de la categoría de ala giratoria se puede definir un helicóptero como una aeronave de alas rotatorias, más pesada que el aire, cuyo rotor gira por la acción de una fuente de potencia produciendo su desplazamiento y su sustentación. Estos, a su vez, se pueden diferenciar por el número de rotores y su colocación (tándem, coaxiales, lado a lado) [28].

La gran ventaja de los helicópteros reside en su capacidad para generar sustentación sin que la aeronave se desplace, permitiendo por tanto despegar y aterrizar de manera vertical (VTOL - Vertical Take-Off and Landing).

En los últimos años la mayor parte del desarrollo de helicópteros no tripulados se ha centrado en su variante de cuatro rotores dispuestos en las extremidades de una cruz. La gran popularidad de este tipo de vehículo conocido como cuadricóptero (figura 2.1 (a)) se debe a su gran maniobrabilidad y relativa facilidad de control.

A diferencia del cuadricóptero, los helicópteros coaxiales (figura 2.1 (b)) son aeronaves con dos rotores que giran en sentidos opuestos sobre un mismo eje. Esta característica le permite consumir menos potencia para una carga dada y ser cons-



Figura 2.1: Helicópteros no tripulados

truido de manera más compacta en comparación con un cuadricóptero. No obstante tiene la desventaja de ser más difícil de controlar, debido a la complejidad del eje principal que tiene que entregar propulsión y control de forma simultánea a dos rotores que giran en sentidos opuestos.

2.2. Helicóptero coaxial

La particularidad del helicóptero coaxial se fundamenta en el uso de dos rotores en contra rotación que compensan y anulan el par resultante sobre el fuselaje de la aeronave. Esto le permite poder prescindir del rotor de cola o anti-par necesario en los helicópteros convencionales para contrarrestar el par generado por el rotor principal. Al tener dos rotores principales, toda la potencia se emplea en generar sustentación aumentando así la eficiencia energética de estos vehículos (figura 2.2). En ensayos experimentales se ha demostrado que con un diseño coaxial se consume un 5 % menos de potencia en vuelo estático que con el mismo empuje en un helicóptero con un sólo rotor principal [11].

Al no necesitar un rotor de cola se consigue una estructura más compacta pudiendo reducir el tamaño del helicóptero un 35-40 % en comparación con un helicóptero convencional. Esto implica a su vez una reducción del momento de inercia del vehículo mejorando así su controlabilidad y manejabilidad [23].

Otra desventaja de los helicópteros convencionales que se evita con un diseño coaxial es la debida a la asimetría de la velocidad. En vuelo de avance, el asa de la hélice que se mueve en el sentido de avance del helicóptero viaja a mayor velocidad a través del aire que el asa que retrocede, provocando así dicha asimetría de la sustentación. En un helicóptero coaxial este efecto se cancela notoriamente debido a los dos rotores principales girando en sentidos opuestos [24].

Como principal inconveniente se tiene la necesidad de incorporar un mecanismo para controlar la aeronave. Al contrario que los cuadricópteros donde el alabeo y el cabeceo se obtienen aumentando y disminuyendo las velocidades de sus rotores, un helicóptero requiere un dispositivo auxiliar para realizar dichas maniobras. Como

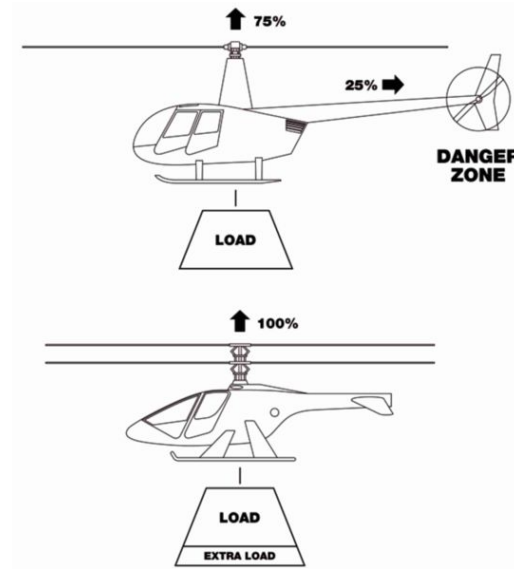


Figura 2.2: Comparación helicóptero convencional con helicóptero coaxial

consecuencia suele ser imprescindible añadir dos actuadores adicionales.

2.3. Configuración, manejo y control

Una elección de diseño que se debe realizar a la hora de desarrollar una aeronave de ala giratoria, a parte del número de rotores y su configuración, es elegir el mecanismo para su manejo. En los helicópteros convencionales el movimiento se consigue variando el ángulo de ataque de las palas del rotor según su posición incrementando así la sustentación creada por la pala. Esta característica, aunque eficaz, resulta compleja de fabricar debido al número de piezas necesarias para su mecanismo. También existen helicópteros que modifican la dirección del flujo de aire a través de las hélices mediante unos timones utilizando un principio similar al de un hidrodreslizador.

En este trabajo se pretende diseñar un mecanismo capaz de mover el centro de gravedad de la aeronave, provocando así que esta se incline y como consecuencia que también incline las hélices. Esto provoca la variación del ángulo del empuje de los rotores y a su vez el desplazamiento del helicóptero.

El principal desafío consiste en diseñar un mecanismo capaz de desplazar el centro de gravedad de manera precisa y rápida, teniendo presente las fuerzas y pares reactivos causados por dicho movimiento que pueden tener un efecto negativo sobre la dinámica del helicóptero.

Un mecanismo, diferente a los mencionados anteriormente, es el desarrollado en [22]. En este artículo se diseña un micro helicóptero coaxial que prescinde de cualquier actuador adicional para efectuar el control de actitud de la aeronave. El movimiento del helicóptero se consigue variando de forma cíclica el ángulo de ataque de las aspas. Pero al contrario que en los helicópteros convencionales donde se utilizan dos servomotores y un plato oscilante, en este trabajo se diseña un mecanismo que varía el ángulo de forma pasiva. Las aspas se sujetan a los motores mediante unas bisagras permitiendo de esta forma a los álabes cambiar su ángulo de incidencia

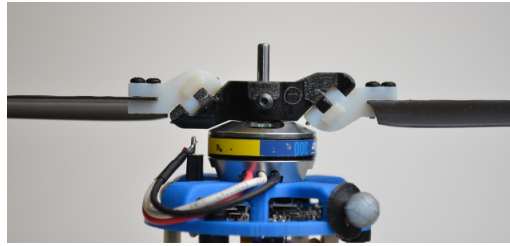


Figura 2.3: Montaje de las aspas con bisagras



Figura 2.4: Plato oscilante del helicóptero Lama

dependiendo del par generado por el motor (figura 2.3). Así el diseño mecánico se simplifica aumentando como consecuencia la dificultad de la computación de bajo nivel para gobernar los motores. Al aumentar el par generado por el motor en una sección de su rotación se incrementa el empuje creado por una de sus palas a la vez que se disminuye el empuje de la otra pala. Este aspecto viene determinado por la orientación de las bisagras permitiendo a una de ellas sólo incrementar su ángulo de incidencia y a la segunda únicamente reducir este ángulo.

2.4. Proyectos relacionados

Se presenta en este apartado la literatura relacionada con el análisis de estas aeronaves.

En el artículo [10] se realiza un modelo dinámico del helicóptero coaxial Lama y se compara a continuación con el modelo de un cuadricóptero para discutir las diferencias entre la dinámica y el control de ambos vehículos. El helicóptero Lama se distribuye como aeromodelismo de ocio y es controlado por radio. Para maniobrar utiliza un plato oscilante parecido al de los helicópteros convencionales, accionado por servomotores (figura 2.4). También dispone de un giroscopio para proporcionar realimentación en lazo de control de la guiñada.

Un proyecto muy similar al presente trabajo se expone en el artículo [9]. Allí se presentan el diseño y el control de un helicóptero coaxial con un mecanismo de desplazamiento del centro de gravedad. El diseño preliminar es capaz de estabilizarse de forma pasiva en el alabeo y cabeceo frente a pequeñas perturbaciones. Cuenta también con un control activo de la altura así como de la guiñada. Para conseguir

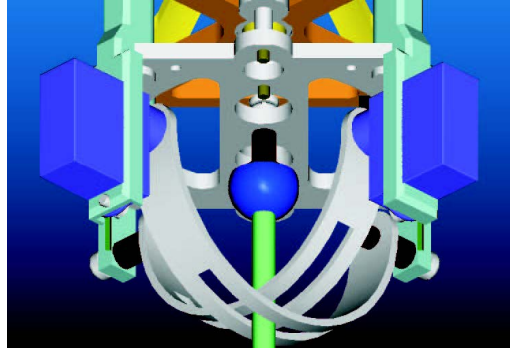


Figura 2.5: Mecanismo de desplazamiento del centro de gravedad



Figura 2.6: Helicóptero micro UAV

el movimiento horizontal de la aeronave mueve el peso de la batería mediante el mecanismo mostrado en la ilustración (figura 2.5). Como se puede observar, se utilizan dos servomotores y una articulación esférica. Al final del escrito se indica que la aeronave es capaz de mantener vuelo estático de sustentación durante periodos cortos de tiempo. En cambio no es capaz aún de maniobrar en el aire.

En el trabajo [8] se describe la construcción de otro helicóptero coaxial detallando el control realizado en el alabeo y cabeceo. El helicóptero diseñado pertenece a la categoría de micro UAVs debido a su pequeño tamaño (figura 2.6). Se incorporan en él sensores infrarrojos para la estimación de altitud así como giroscopios e inclinómetros para calcular la posición de la aeronave. Los reguladores utilizados para los controles son del tipo proporcional derivativo (PD).

El desarrollo de un modelo matemático de la dinámica de estos helicópteros se explica en [25]. Una vez creado el modelo dinámico se realizan simulaciones que demuestran la viabilidad de las aeronaves frente a diferentes condiciones. También se implementa un control de altura. La tesis concluye con la presentación de diversos prototipos (figura 2.7), pero indica que ninguno es capaz de realizar un vuelo autónomo, debido a la falta de fiabilidad del mecanismo de desplazamiento del centro de gravedad y el deterioro prematuro de este después de los primeros ensayos.

El proyecto de investigación Gimball [13], ejecutado durante cinco años en la Escuela Politécnica Federal de Lausana en Suiza, ha desarrollado un helicóptero

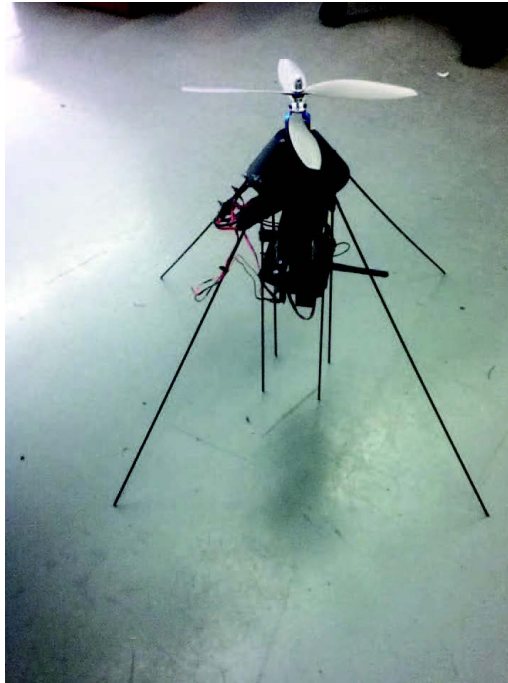


Figura 2.7: Prototipo helicóptero coaxial con desplazamiento del centro de gravedad

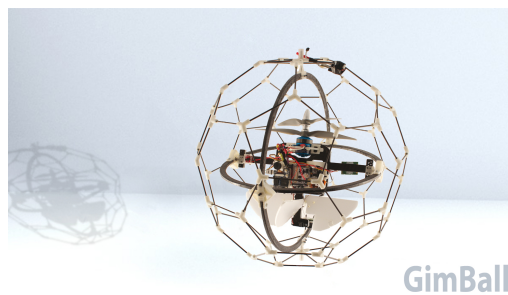


Figura 2.8: Helicóptero Gimball de Flyability

coaxial controlado mediante la desviación del flujo de aire de sus rotores (figura 2.8). Utiliza para ello dos alerones dispuestos en forma de cruz debajo de las hélices. La particularidad de este diseño se encuentra en su protección. Una estructura esférica envuelve la aeronave y la permite sufrir colisiones sin por ello perder la estabilidad. Este vehículo de Flyability es capaz de acceder a terrenos accidentados o construcciones complejas pudiendo desempeñar labores de seguridad y rescate. Para permitir al operario, encargado de su vuelo, poder realizar las maniobras dispone de una cámara para visualizar la perspectiva del helicóptero.

Flone [6] es una plataforma que monta un teléfono móvil en un cuadricóptero (figura 2.9). Este proyecto propone la incorporación del teléfono con la expectativa de que se desarrollen diferentes aplicaciones para aumentar así su funcionalidad. Actualmente se encuentra en proceso de desarrollo. En las primeras etapas dispone de un controlador de vuelo Multiwii basado en Arduino y una aplicación de grabación de vídeo para el teléfono móvil. Se pretende en etapas posteriores prescindir del microcontrolador y así simplificar el diseño.



Figura 2.9: Plataforma Flone para teléfonos móviles

Capítulo 3

Componentes del helicóptero

En este apartado se describen todas las partes físicas que componen el helicóptero.

A la hora de diseñar un primer prototipo, con las características descritas anteriormente, el primer paso es diseñar el mecanismo encargado de desplazar el centro de gravedad de la aeronave. Para ello es necesario conocer tanto partes físicas que se pretenden trasladar, como el movimiento de dicha traslación y por último el actuador encargado de llevarla a cabo.

Los requisitos impuestos para la aeronave son los de diseñar un helicóptero capaz de realizar vuelos en interiores y exteriores. A su vez, la incorporación de un teléfono móvil en su estructura condiciona el tamaño del helicóptero con un peso estimado de un kilogramo.

3.1. Motores

Como parte fundamental en un diseño coaxial, se necesitan dos motores capaces de girar dos hélices sobre un mismo eje en sentidos opuestos (figura 3.1). En la práctica del aeromodelismo se utilizan para ello motores eléctricos sin escobillas. Estos motores prescinden de las escobillas utilizadas para realizar el cambio de polaridad del rotor reduciendo así el rozamiento creado por estas y aumentando el rendimiento. Como consecuencia necesitan una señal trifásica para crear la rotación del campo magnético. En el caso que la alimentación se realice mediante corriente continua se necesita utilizar un controlador electrónico que genere una señal trifásica modulada en ancho de pulso (PWM).

Los motores con rotores coaxiales son además de tipo *outrunner*. En estos motores la parte móvil es la carcasa exterior donde van montados los imanes permanentes. La parte fija es la interior y es donde se encuentran los bobinados de hilo conductor. Además este tipo de motores disponen de sensores que detectan la posición del rotor para que el controlador electrónico pueda activar y desactivar la bobinas en el momento preciso.

Un parámetro muy importante en los motores sin escobillas es el factor kV . Este valor nos indica la velocidad angular en revoluciones por minuto a la que puede girar el motor, sin carga, para un valor de tensión dado en voltios.

Por último, otro factor determinante a la hora de escoger un motor para un helicóptero es la cantidad de empuje que puede generar. Este valor sólo se conoce para un conjunto de motor y hélices siendo necesario decidir el tamaño y el ángulo de ataque de estas.



Figura 3.1: Motor sin escobillas CR28M



Figura 3.2: Batería LiPo Rhino 4900mAh

Una vez conocidas las hélices que montan los rotores se puede conocer la potencia y corriente consumida.

3.2. Batería

Las baterías son dispositivos que convierten la energía química almacenada en celdas electroquímicas en energía eléctrica. En la actualidad se utilizan baterías de polímero de litio (LiPo) (figura 3.2) en el modelismo de radiocontrol, así como en los vehículos aéreos no tripulados de menores dimensiones. Estas baterías permiten una mayor densidad de carga y una gran tasa de descarga necesaria para alimentar motores como los descritos anteriormente y ser montadas en vehículos aéreos. Estas características se pueden observar en la comparación con varios tipos de baterías realizado en la siguiente tabla (tabla 3.1).

Como se advierte en la gráfica (figura 3.3), las baterías de polímero de litio presentan la mayor relación de potencia por peso. De este modo, para un mismo peso, una batería de polímero de litio puede entregar más energía que el resto. Esta característica resulta muy significativa en aeronaves, donde el peso de cada componente es determinante.

Cada celda de la batería tiene un voltaje nominal de 3,7V y un voltaje mínimo de 3 V. Este último valor es muy importante debido a que la batería queda permanentemente dañada si se descarga por debajo de su tensión mínima.

Tabla 3.1: Comparación entre diferentes tipos de baterías

Tipo	Plomo	Níquel-Cadmio	Níquel-Hidruro	Polímero de Litio
Voltaje por célula	2 V	1,2 V	1,2 V	3,7 V
Potencia/Peso	30 Wh/kg	50 Wh/kg	70 Wh/kg	100-130 Wh/kg
Número de recargas	~ 1000	~ 500	~ 1000	~ 5000
Tasa descarga/mes	5 %	30 %	20 %	6 %

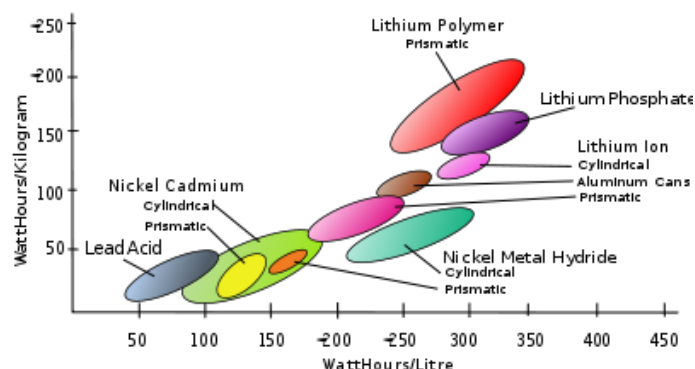


Figura 3.3: Densidad de carga de los diferentes tipos de baterías

En las baterías de polímero de litio se utiliza la nomenclatura $xSyP$, donde x representa el número de celdas en serie e y indica las que se encuentran en paralelo.

El principal inconveniente de este tipo de baterías es que deben ser manipuladas con cuidado debido a que pueden llegar a explotar si se realiza un uso inadecuado con ellas.

3.3. Controladores electrónicos

Los controladores electrónicos llamados también ESC por sus siglas en inglés (Electronic Speed Controller) son circuitos electrónicos encargados de gobernar la velocidad de los motores eléctricos (figura 3.4). En los vehículos de radiocontrol con motores sin escobillas generan una señal trifásica modulada en ancho de pulso. Como señal de entrada reciben la misma señal que los servomotores a $50Hz$, variando el ancho de pulso de $1ms$ hasta $2ms$ para la máxima velocidad.

La mayoría de variadores cuentan también con un circuito denominado BEC (Battery Eliminator Circuit). Como estos controladores son alimentados a través de una batería, aprovechan los terminales de alimentación del conector tipo servo para alimentar a los posibles servomotores conectados en paralelo.

3.4. Hélices

Las hélices tienen principalmente tres características. En primer lugar, aunque lo más habitual son las hélices con dos palas, también se pueden encontrar variantes con tres o más álabes. A parte cada hélice se define con dos valores expresados en pulgadas. El primero de ellos es el diámetro del círculo que trazan las puntas de las



Figura 3.4: Controlador electrónico Turnigy Plush 25A



Figura 3.5: Servomotor HD-3001HB

hélices al rotar. El segundo se refiere al paso. Este expresa el avance de las palas con cada giro suponiendo que se encontraran en un medio sólido.

3.5. Servomotores

Un servomotor o servo es un dispositivo actuador capaz de rotar dentro de un rango de operación (figura 3.5). Está formado por un motor de corriente continua con un mecanismo reductor y un circuito de control con un potenciómetro para poder controlar su posición. Para controlar el servomotor se envía una señal modulada en ancho de pulso (PWM) con una frecuencia de 50 Hz. La anchura del pulso codifica el ángulo de giro. Esta varía en un servo estándar, con un rango de movimiento de 180 grados, entre $1ms$ y $2ms$.

Los servomotores tienen tres terminales de conexión igual que los controladores electrónicos descritos anteriormente. Siendo dos terminales para la alimentación eléctrica del circuito y el tercero para la entrada de la señal de control.

MOTORES HÉLICES									
Modelo	Current (A)	Max Current (A)	Power (W)	Propellers	Peso (g)	Trhust máximo (g)	ESC	Precio	
CR23L Contra Rotating BL System1100kv	17,7	20,02	254	GWS 9x5	109	960	2 x 25 A	\$	41,99
CR23M Contra Rotating BL System1550kv	18	21,04	269	GWS 8x4	73	800	2x 25 A	\$	42,99
CR23S Contra Rotating BL System 1700kv	19	21	264	GWS 8x5	64	744	2x 25 A	\$	43,55
Contra Rotating BL System 375W	21	34	375	10x4 or 10x5	79	1320	2 x 25 A	\$	59,99

Figura 3.6: Oferta motores coaxiales

3.6. Elección de componentes

Como se deduce de la descripción de los componentes, la elección de cada uno de ellos viene determinada en gran medida tanto por el diseño final del helicóptero como por la necesidad de ser compatibles entre si. En primer lugar, la elección del motor junto con la pareja de hélices depende directamente de la masa que necesiten levantar. Para ello lo primero que se evalúa es el peso total del helicóptero, escogiendo unos valores medios para las masas de todos los componentes necesarios y estimando el posible peso de la estructura de la aeronave. Este procedimiento se efectúa de forma iterativa pudiendo realizar mejores estimaciones a medida que se van precisando las características necesarias para los componentes y se concretan sus masas.

A continuación se muestran los valores medios utilizados para los pesos de los componentes (tabla 3.2).

Tabla 3.2: Pesos medios de los componentes

Componente	Peso
Motores y hélices	80g
Batería	325g
Variadores electrónicos	2x20g
Servomotores	2x30g
Microcontrolador	20g
Teléfono móvil	150g
Peso total componentes	675g

A parte de evaluar el posible peso de la estructura del helicóptero es necesario establecer un margen para el empuje de los motores que permita al helicóptero maniobrar con agilidad.

Si se observa la oferta de motores coaxiales capaces de levantar el peso calculado, se encuentran cuatro opciones (figura 3.6). Cada una de ellas tienen respectivamente empujes de 744, 800, 960 y 1320 gramos. Las dos primeras alternativas no ofrecen suficiente potencia y la tercera limita un excedente de peso a 285 gramos. Esta diferencia restringe demasiado el diseño de la estructura, escogiéndose finalmente el último motor.

Este motor (CR28M) tiene unas hélices instaladas de dimensiones 10x6 logrando así el empuje de 1320g con una potencia de 375W (tabla 3.3).

Con esta elección la batería necesaria para una duración de vuelo superior a 10 min necesita una capacidad superior a 3500mAh. Este valor viene determinado por la intensidad nominal de los motores de 21A:

Tabla 3.3: Especificaciones motor CR28M

Kv	1050
Intensidad nominal	21 A
Potencia máx.	375 W
Intensidad máx.	34 A
Empuje máx.	1320 g
Dimensiones	28 x 51 mm
Peso	79 g

BATERÍAS RHINO							
Modelo	Capacidad (mAh)	Discharge	I max. (C*mAh) [A]	t max. (mAh/21A) [min]	Peso (g)	Precio	
R2350-20-3	2350	20	47	6,71	205	\$	14,00
R2550-25-3	2550	25	63,75	7,29	190	\$	12,55
R2550-40-3	2550	40	102	7,29	231	\$	18,00
R3700-20-3	3700	20	74	10,57	330	\$	16,49
R3700-25-3	3700	25	92,5	10,57	327	\$	19,95
R4000-25-3	4000	25	100	11,43	342	\$	23,40
R4900-20-3	4900	20	98	14,00	352	\$	22,49
R4900-25-3	4900	25	122,5	14,00	395	\$	37,13

Figura 3.7: Oferta baterías 3S1P

$$Capacidad[mAh] = \frac{10min}{60\frac{min}{h}} * 21A * \frac{1000mA}{1A} = 3500mAh$$

Además, para cumplir con las necesidades del motor necesita ser del tipo 3S1P (11,1 V) y tener una constante de descarga suficiente para poder entregar intensidades de 34A (figura 3.7). La batería Rhino 4900mAh 3S1P 20C supera algunas de las necesidades descritas aumentando sólo levemente el peso siendo finalmente la elegida (tabla 3.4).

Las especificaciones del motor indican como requerimiento dos controladores de 25A con lo que se eligen dos Turnigy Plush 25amp Speed Controller (tabla 3.5) debido a que toleran las intensidades máximas solicitadas por los motores (figura 3.8).

Para desplazar el centro de gravedad se eligen dos servomotores Standard HD - 3001HB con un par máximo de 3,5 kg·cm a 4,4 V, suficientes para dicho fin (tabla 3.6).

3.7. Arduino®

Arduino es una plataforma de hardware libre, basada en una placa con un micro-controlador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica

Tabla 3.4: Especificaciones batería Rhino 4900mAh 3S1P

Capacidad	4900 mAh
Constante de descarga	20 C
Configuración	3S / 11.1 V
Dimensiones	165 x 25 x 41 mm
Peso	352 g

ESC							
Modelo	Peso (g)	Tamaño (mm)	Current (A)	Burst (A)	BEC(A)		Precio
HK-SS30A-HW	24	45x25x4	25	30	3	\$	9,98
TURNIGY Plush 25amp	22	24x45x11	25	35	2	\$	13,45

Figura 3.8: Oferta variadores electrónicos de 25 A

Tabla 3.5: Especificaciones controladores electrónicos Turnigy Plush 25 A

Intensidad nominal	25 A
Intensidad máxima	35 A
BEC	5 V / 2 A
Celdas LiPo	2 - 4
Dimensiones	24 x 45 x 11 mm
Peso	22 g

en prototipos de proyectos multidisciplinarios [1]. Su hardware consiste principalmente en un microprocesador Atmel AVR y puertos de entrada y salida tanto digitales como analógicos para la conexión de sensores y actuadores [2].

El microcontrolador en este proyecto es el encargado de recibir los datos de los sensores y los comandos de manejo del teléfono, que va montado en el helicóptero, y con ellos controlar la aeronave. Los requisitos mínimos son por lo tanto contar con cuatro canales de salida con modulación por ancho de pulso, para controlar dos servomotores y dos variadores de los motores, y tecnología para poder comunicarse con el teléfono móvil.

Para este último aspecto existen dos opciones inalámbricas posibles. Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de datos entre diferentes dispositivos mediante enlace por radiofrecuencia. Está diseñado especialmente para dispositivos de bajo consumo y que requieran corto alcance de emisión. El alcance varía dependiendo de la potencia de transmisión con 10 m para potencias de 2,5 mW (Clase 2) o 100 m para potencias de 100 mW (Clase1). La segunda tecnología también basada en la transmisión de datos inalámbrica por radiofrecuencia es la conocida como WiFi o su estándar IEEE 802.11. Tiene un alcance de 200 m y una velocidad máxima de 11 Mbps en el estándar 802.11b, 100 m y 54 Mbps en el estándar 802.11g y 250 m de alcance a una velocidad de 108 Mbps en la última versión 802.11n.

Para facilitar la comunicación con el teléfono se elige una placa Arduino Yún (figura 3.9 (a)). Esta placa combina Arduino con una distribución de Linux llamada Linino, específica para Arduino. Por un lado se integra un sistema Arduino basado en el procesador Atmega32U4 y por otro un procesador con una distribución basada en OpenWRT con conectividad WiFi y con Linino instalado. Dispone también con

Tabla 3.6: Especificaciones servomotor HD - 3001HB

Velocidad	0,15 s/60°
Par máx.	3,5 kg/cm
Dimensiones	40,7 x 20,5 x 39,5 mm
Peso	43 g

Tabla 3.7: Especificaciones Arduino Yún

Microcontrolador	ATmega32u4
Voltaje entrada	5 V
Pines digitales E/S	20
Pines digitales E/S PWM	7
Pines analógicos entrada	12
Frecuencia reloj	16 MHz
Memoria Flash	32 kB
WiFi	IEEE 802.11 b/g/n
Dimensiones	53 x 73 mm
Peso	32 g

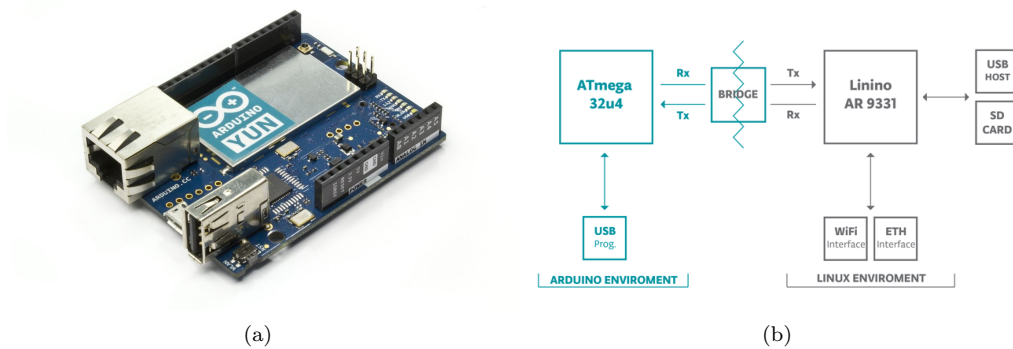


Figura 3.9: Arduino Yún y librería Bridge

los pines necesarios de entrada y salida (tabla 3.7).

Una particularidad de esta placa es la librería Bridge que facilita la comunicación entre los dos procesadores permitiendo que los programas de Arduino ejecuten comandos en Linino y que acceda a los periféricos conectados al procesador AR 9331 (figura 3.9 (b)).

3.8. Zum BT-328

A lo largo del proyecto, después de la fabricación de la placa *shield* en el capítulo de Pruebas y Prototipos (sección 4.6), surge la posibilidad de cambiar la placa Arduino Yún por una Placa ZUM BT-328 (figura 3.10). Esta placa de la empresa BQ es completamente compatible con Arduino®. Como principales características (tabla 3.8), relevantes para este proyecto, podemos encontrar pines macho para el conexionado directo de conectores servo y tecnología Bluetooth incorporada. Además permite una alimentación de entrada de 5,8 V hasta 16 V lo que hace posible que se alimente directamente desde la batería de 11,1 V evitando así la necesidad de utilizar un regulador lineal en el helicóptero.

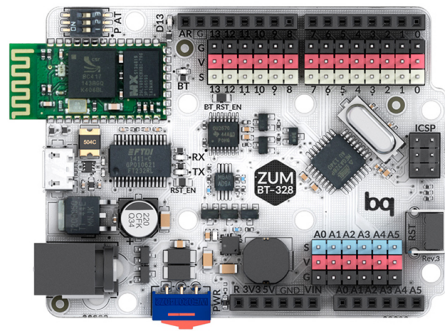


Figura 3.10: Placa ZUM BT-328

Tabla 3.8: Especificaciones BQ ZUM BT-328

Voltaje entrada	6 - 17 V
Pines digitales E/S	14
Pines analógicos entrada	6
Bluetooth	2.1
Dimensiones	67 x 54 mm

Capítulo 4

Pruebas y prototipos

Con los componentes seleccionados es conveniente probar su funcionamiento en conjunto para poder identificar posibles defectos o interferencias entre ellos. Esta prueba sirve además para familiarizarse con los componentes.

Por ello en primer lugar se diseña una estructura a modo de banco de pruebas. Para su fabricación se elige utilizar piezas de polimetilmetacrilato, conocido comúnmente como metacrilato. Este plástico transparente en un 93 % destaca principalmente por su alta resistencia al impacto y una densidad de tan sólo $1190 \frac{kg}{m^3}$. El metacrilato se comercializa en planchas rectangulares de diferentes grosores.

Además se fabrica una carcasa para la batería con la finalidad de que pueda ser desplazado su peso por los servos. En este caso se utiliza el termoplástico ABS (acrilonitrilo butadieno estireno), también muy resistente a los impactos e imprimible en 3D.

4.1. Fresado CNC

Para la elaboración de la estructura se diseña en primer lugar un modelo en 2D con AutoCAD 2015. Este software de Diseño Asistido por Computadora o CAD, por sus siglas en inglés, es desarrollado y comercializado por la empresa Autodesk. Una vez diseñadas las piezas que forman la estructura (figura 4.1) se exportan en formato DXF (acrónimo del inglés Drawing Exchange Format) para que puedan ser interpretadas por el software necesario para su posterior fresado. Este software se compone en este caso de dos programas. En primer lugar se utiliza VCarve para calcular las trayectorias de fresado y taladro para el mecanizado en la máquina de control numérico. Una de las características más importantes de este programa es la posibilidad de visualizar en la pantalla del ordenador el resultado del diseño. Esta propiedad permite reconocer problemas de diseño o configuración de las trayectorias con rapidez. La forma de trabajar con VCarve es muy intuitiva. En primer lugar se seleccionan las características del material de trabajo definiéndose el área disponible y su grosor. A continuación se coloca sobre este área el dibujo de la pieza exportado por AutoCAD. Por último se van seleccionando contornos cerrados de dicho dibujo y junto con la selección de un método de mecanizado así como de la herramienta a utilizar el programa calcula la trayectoria necesaria. En este momento conviene visualizar la trayectoria creada para confirmar que cumple con todos los requisitos y una vez aprobado se pueden exportar las trayectorias en un archivo TXT.

La segunda herramienta, Match 5, se utiliza para controlar la máquina fresadora

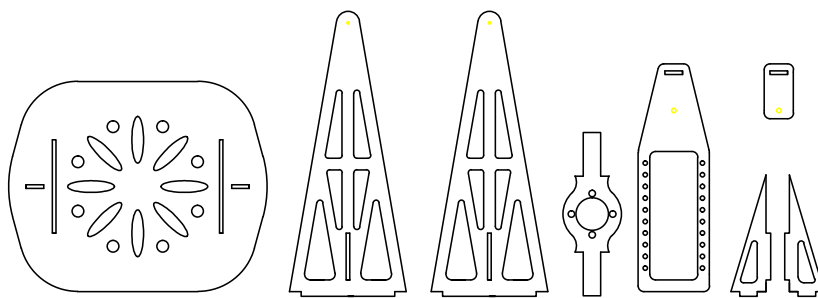


Figura 4.1: Diseño piezas en AutoCAD

por Control Numérico Computarizado (CNC). Este programa utiliza las trayectorias de herramienta definidas anteriormente para gobernar la fresadora. Cuenta principalmente con una botonera virtual de puesta en marcha, parada y un pulsador de emergencia, así como con teclas para maniobrar manualmente la fresadora. Para empezar a fresar es importante en primer lugar sujetar firmemente la plancha de polimetilmetacrilato sobre la mesa de la máquina de mecanizado. Un método muy cómodo y efectivo para tal fin es utilizar cinta de doble cara asegurándose que la zona de fresado quede fuertemente asegurada. Esto evita que durante el fresado puedan desprenderse trozos de metacrilato que puedan herir o lesionar a personas. A continuación se coloca la fresa escogida en el mandril. En este caso se trata de una fresa universal de 3mm de carburo de tungsteno (*carbide*). Con la plancha fija y la fresa en su lugar, el siguiente paso es indicar al programa Match 5 las coordenadas que tomará de origen. Para ello se mueve el cabezal a la zona de metacrilato escogida para el fresado de la pieza. Es muy importante ajustar lo mejor posible la altura de la fresa debido a que será la encargada de la profundidad de fresado y taladro. Si no es la correcta puede no taladrar el grosor completo del material. En este momento es posible empezar el fresado, siendo buena costumbre mantenerse cerca del pulsador de emergencia y no despreocuparse del mecanizado.

4.2. Impresión 3D

Como la carcasa de la batería no necesita propiedades estructurales importantes, se decide utilizar tecnología de impresión en tres dimensiones para su creación (figura 4.2). Estas máquinas crean piezas volumétricas a partir de diseños por ordenador. En este caso la impresora es aditiva de inyección de polímero depositando el material por capas según avanza la impresión. Su funcionamiento se basa en un cabezal extrusor con tres grados de libertad en las dimensiones XYZ y un software que secciona un modelo 3D en capas para su impresión. Esta tecnología permite realizar modelos con formas complejas de forma relativamente sencilla.

Igual que con el fresado por control numérico el primer paso consiste en crear un diseño CAD (figura 4.3). En este caso al necesitar un modelo en tres dimensiones el software utilizado es Autodesk Inventor Professional 2015. Este se basa en técnicas de modelado paramétrico permitiendo modelar la geometría, dimensiones y material de las piezas. Tiene además una gran ventaja y es la creación de ensamblajes a



Figura 4.2: Impresora 3D Prusa i3

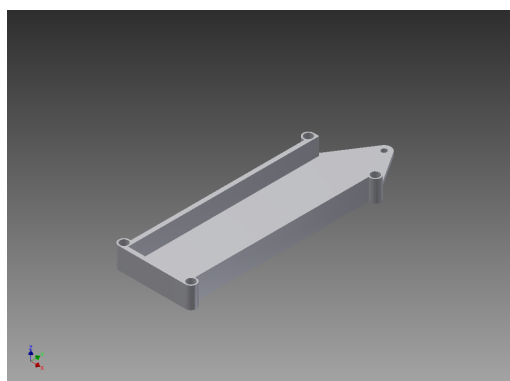


Figura 4.3: Diseño carcasa para la batería

partir de diferentes piezas. Las piezas son ensambladas agregando restricciones lo que permite visualizar, antes de ser fabricadas, si encajarán en el diseño final o en caso contrario si no se acoplan bien y deben ser modificadas o desechadas. Es por esta cualidad que incluso los diseños en metacrilato, que no necesitan modelos en 3D, conviene realizarlos asimismo con este software. Así se pueden realizar los ensamblajes incluyendo el resto de componentes, como pueden ser en este caso los servos, para verificar que las dimensiones escogidas son apropiadas.

Una vez finalizado el diseño de la pieza y comprobado su viabilidad se debe exportar en formato STL (STereo Litography). Este formato define la geometría de los objetos en tres dimensiones excluyendo información no relevante para el prototipado como el color, texturas o propiedades físicas que sí incluyen otros formatos de diseño asistido por computador. Los archivos en este formato se abren a continuación con el programa Slic3r. Esta herramienta convierte el modelo digital en tres dimensiones en instrucciones de impresión. Para ello secciona el modelo en capas, generando las trayectorias para rellenarlas y calculando la cantidad de material a extruir. Entre las diferentes propiedades que se pueden seleccionar para esta tarea se encuentran la elección del grosor de las capas y la densidad de relleno de estas. El archivo creado por Slic3r, llamado G-code, se abre a continuación con Pronterface. Este progra-

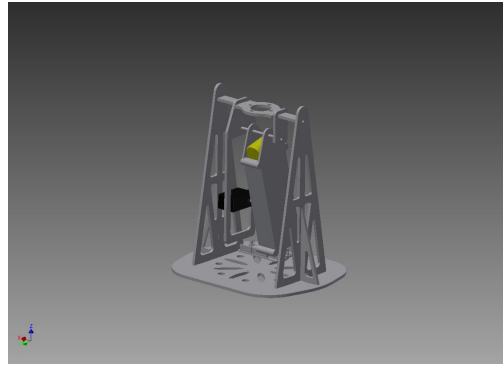


Figura 4.4: Diseño del primer banco de pruebas

ma es el encargado de comunicarse y manejar la impresora Prusa i3 a partir del código en lenguaje G. También dispone una interfaz gráfica (GUI) donde el usuario puede controlar el movimiento del cabezal y las temperaturas del extrusor y de la cama de impresión. Cuando las temperaturas alcanzan la temperatura de trabajo, la impresora estará preparada para imprimir.

4.3. Banco de pruebas

Con todas la piezas fabricadas se procede a realizar el montaje del banco de pruebas. Las piezas de metacrilato han sido diseñadas de tal forma que puedan ser montadas sin necesidad de utilizar elementos de fijación. Simplemente es necesario introducir las pestañas en sus respectivas ranuras. Como se puede observar en las figuras (figuras 4.4, 4.5), el banco de pruebas está formado por dos partes bien diferenciadas. Por un lado consta de una base, con sus dos paredes, que es la encargada de sostener al helicóptero. Las paredes están reforzadas estructuralmente con dos contrafuertes debido a su gran altura en comparación con su estrecho grosor, lo que las hace ser demasiado flexibles para la aplicación necesitada. La función principal de la base es la de permitir el giro del helicóptero debido al desplazamiento del centro de gravedad pero impidiendo que pueda volcar o despegar. Al añadir ruedas a la base se permite a su vez que esta pueda desplazarse sobre las dos dimensiones del suelo. La segunda parte del banco de pruebas es la relativa al helicóptero. En este primer prototipo simplificado cuenta sólo con los motores, la batería y un servo para desplazarla. Esta configuración permite verificar el correcto funcionamiento de los rotores y conocer el giro que realiza el motor como reacción al movimiento de la batería. Como no se conoce el lugar óptimo para colocar el servomotor, existe un raíl con diversos anclajes dónde se puede ir cambiando su posición. El eje que permite el movimiento entre la base y el helicóptero está formado por unos alfileres encajados en unos taladros realizados longitudinalmente en el metacrilato.

En las primeras pruebas realizadas se encuentran diversos defectos en el diseño de la estructura como la falta general de rigidez.

Otro punto objeto de mejora es el eje que permite la rotación del helicóptero. A parte de su debilidad, este se encuentra alejado del centro de gravedad de la futura aeronave. Este aspecto impide que el movimiento observable en el banco de pruebas se asemeje al del futuro helicóptero en vuelo. Por ello se decide realizar un segundo diseño.

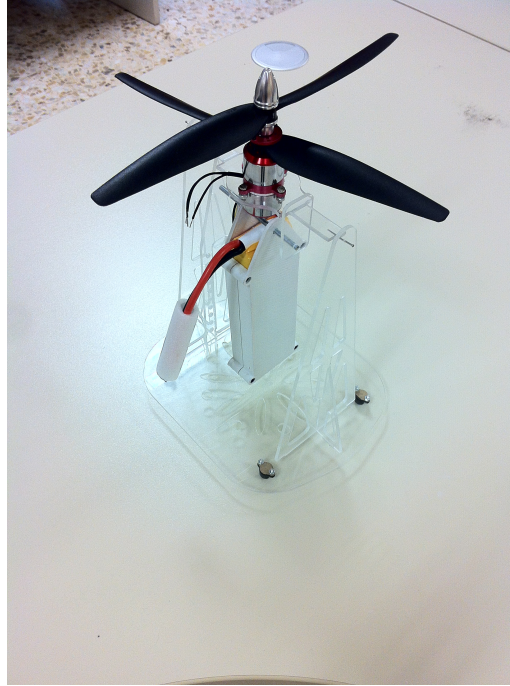


Figura 4.5: Primer banco de pruebas

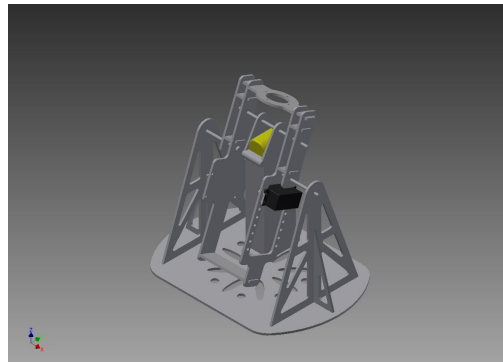


Figura 4.6: Segundo banco de pruebas

En la figura (figura 4.6) se contempla como se ha desplazado el eje sobre el centro de gravedad. Además se ha reforzado la estructura del helicóptero y cambiado el eje de alfileres por unos tornillos de métrica M4. Al descender el eje de giro se ha podido recortar la altura de las paredes de la base consiguiendo así que también aumente su robustez. Por último se aprovecha la creación de un nuevo prototipo para añadir sujeciones dónde colocar la placa de Arduino y los controladores de los motores.

4.4. Conexión de los componentes

Tanto los variadores electrónicos, como la batería y los motores comprados no disponen de conectores para realizar las conexiones. Por ello es necesario elegir unos conectores apropiados y soldarlos. En este caso se necesitan dos conectores de oro de 6mm para la alimentación de la batería y seis conectores de tipo banana de 2mm para las señales trifásicas de ambos motores. La batería se conecta en la entrada de

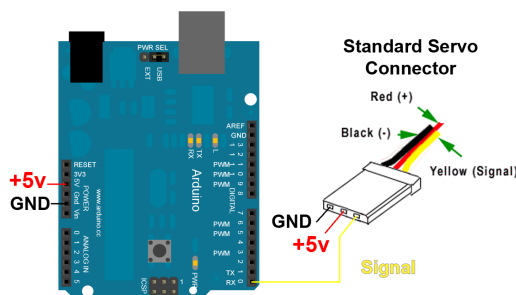


Figura 4.7: Conexión de conector tipo servo

los dos controladores electrónicos y los tres cables de salida de estos variadores se conectan a cada uno de los motores.

Debido a la falta de marcas o señales a la hora de realizar las conexiones, no se conoce que variador controla cada uno de los motores. Para averiguarlo hay que descubrirlo experimentalmente. Otro de los fenómenos que se pueden dar es a la hora de arrancar los motores si alguno de ellos o los dos giran en sentidos equivocados. En este caso, para solucionarlo, basta con intercambiar entre si dos de las tres fases que transmiten la señal de los controladores a los motores.

El siguiente paso es conectar los conectores de servo tanto de los variadores como del servomotor a la alimentación y la placa Arduino (figura 4.7). Para facilitar esta tarea se utiliza una placa de prototipado. Los conectores de tipo servo tienen tres cables. El de color rojo es para la alimentación positiva, siendo habitual 5V, el negro para la referencia de tierra y el último de color naranja o amarillo es el de señal. En el caso de trabajar con una placa Arduino hay que asegurarse que el cable de señal se conecta a uno de los pines digitales preparados para mandar señales por modulación por ancho de pulso.

4.5. Programa Arduino

Un programa creado en el entorno de desarrollo integrado (en inglés Integrated Development Environment; IDE) de Arduino se conoce como *sketch*. El lenguaje con el que se programa es un derivado del lenguaje de programación Wiring con una estructura y sintaxis muy similar a la del lenguaje C. Como partes fundamentales de un *sketch* se encuentran dos bloques. El bloque `setup()` se ejecuta una sola vez cuando comienza el programa y es el encargado de recoger todos los parámetros de configuración. Por otro lado, la parte de código representada como `loop()` es un bucle que se recorre de forma ininterrumpida y es donde se escribe el programa.

Para empezar a controlar los motores y el servo se escribe un programa sencillo en el que se pasan los comandos introducidos por el teclado del ordenador a través del terminal serie de Arduino. Tanto el servo como los motores se controlan con ayuda de la librería `servo.h` y el método `writeMicroseconds()`. Este último permite modular el ancho de pulso de la señal de 50Hz entre 1000μs y 2000μs, pudiendo controlar así la posición en el servo y la velocidad de los motores.

Una vez comprobado el correcto funcionamiento de los componentes así como la facilidad de realizar un programa para manejar el servo y los motores, se pretende aprovechar la tecnología WiFi puesta a disposición por la placa Arduino Yún. Esto anula la necesidad de mantener el controlador conectado directamente a un ordena-

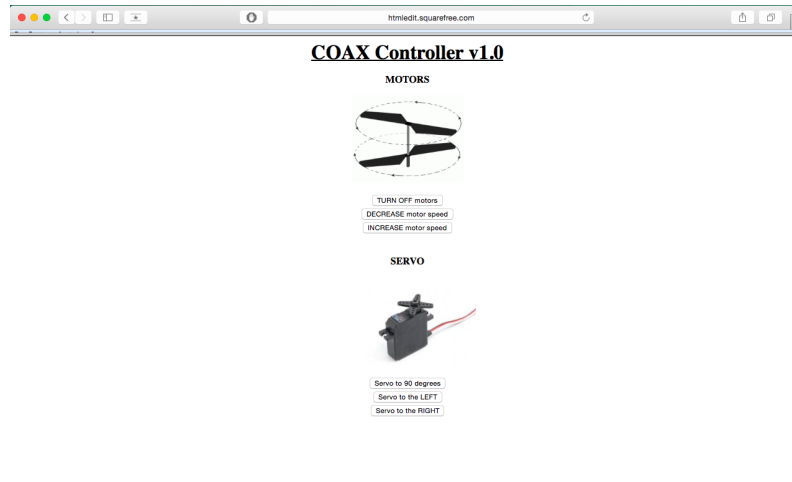


Figura 4.8: Interfaz gráfica en HTML

dor para recibir los comandos de movimiento. Como consecuencia, al ser inalámbrico, permite rodar al banco de pruebas por el suelo sin verse afectado por el arrastre de cables.

El segundo *sketch* crea un servidor HTTP en el microcontrolador. Este programa es el encargado de abrir un puerto de comunicación con los clientes de la aplicación y atender sus peticiones. Para que el cliente, en este caso el operario del helicóptero, pueda enviar comandos de manera sencilla se crea una interfaz gráfica con HTML (figura 4.8). Así, el cliente, sólo tiene que acceder desde su navegador a la dirección `http://yun.local/arduino/` mientras está conectado a la misma red local que la placa Yún para poder manejar el helicóptero. El resto del programa, encargado de enviar las señales a los motores y al servo, se mantiene igual que en el primer *sketch*.

Durante las pruebas se descubre el inconveniente de realizar las conexiones de los terminales servo en la placa de prototipado. El número de cables necesario para realizar todas las conexiones, al ser elevado, resulta engorroso y aumenta la probabilidad que alguno de ellos pueda soltarse o sea colocado incorrectamente, además del tamaño y peso añadido. Por ello se decide diseñar una placa que facilite dicha tarea.

4.6. Placa PCB

Para el diseño de la placa de circuito impreso se utiliza el software EAGLE de la compañía CadSoft Computer. Con él se pueden trazar las pistas de material conductor necesarias para conectar eléctricamente y de manera apropiada las diferentes partes del circuito. En este caso el diseño resulta muy sencillo debido a la ausencia de componentes electrónicos utilizados. Basta con delinear los caminos para conectar cada uno de los cuatro conectores tipo servo necesarios a la alimentación, tierra y sus respectivas señales.

Para que la placa resulte compacta y fácil de instalar en el helicóptero se diseña de forma que pueda ser montada sobre la placa Arduino. Este tipo de placas externas

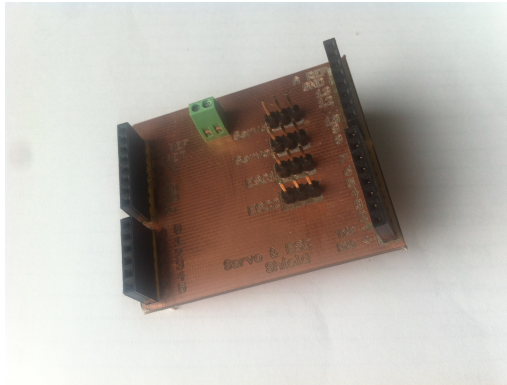


Figura 4.9: Placa PCB

se conocen como *shield* y se introducen en los pines de entrada y salida de la placa Arduino. Para no obstruir el acceso a los pines se extienden estos a un nivel superior.

El diseño terminado se puede visualizar con la herramienta Gerbv. Este programa permite comprobar el correcto trazado de las pistas así como la colocación de los agujeros y el perfil escogido para la placa.

A continuación se fresa la placa y se sueldan los diferentes conectores. Se necesitan cuatro regletas hembra de 2,54mm para extender los pines de la placa Arduino, cuatro pines triples macho de 2,54mm para los conectores tipo servo y una regleta clema con dos contactos para conectar la alimentación (figura 4.9).

4.7. Placa BQ

Como se explica en el apartado de Componentes del helicóptero (capítulo 3), es en este momento cuando aparece la posibilidad de utilizar la placa ZUM BT-328 de la empresa BQ.

El cambio de la tecnología Wifi por Bluetooth no resulta negativo, ya que sólo es necesaria para la comunicación entre la placa y el teléfono móvil montado en el helicóptero. Estos se encuentran a una distancia de unos centímetros no siendo determinante la distancia mínima alcanzada por el Bluetooth.

Por estos motivos, además de las cualidades de la placa ZUM BT-328, se descarta el uso de la placa Arduino Yún en favor de la placa ZUM.

En este momento se modifica el último programa de Arduino para incluir la comunicación por Bluetooth. El nuevo *sketch* es muy parecido al primero en el que la comunicación se hace a través del terminal del ordenador. Esto es debido a que la placa ZUM, cuando está emparejado su módulo Bluetooth con otro dispositivo, se comunica por defecto a través de dicho módulo. Esta característica viene determinada por la multiplexación de la comunicación por el puerto serie tanto Bluetooth como USB. Como consecuencia sólo una de las dos puede utilizarse al mismo tiempo pero tiene la ventaja de poder programarse indistintamente ambos tipos de comunicación.

Para poder manejar el helicóptero aprovechando la tecnología Bluetooth y comprobar el correcto funcionamiento del *sketch* programado, se descarga la aplicación, sin coste, ArduinoRC para teléfonos con sistema operativo Android (figura 4.10).

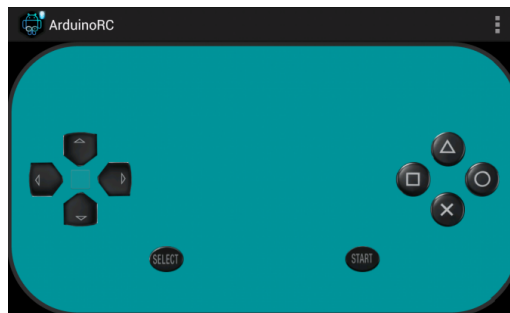


Figura 4.10: Interfaz de la aplicación ArduinoRC

Esta aplicación permite después de emparejar el teléfono con la placa ZUM mandar comandos a través de Bluetooth. Además tiene una interfaz gráfica con los controles necesarios para que el manejo del helicóptero sea más intuitivo.

4.8. Prototipos del helicóptero

Durante los ensayos realizadas con el banco de pruebas se observa el riesgo que suponen las dos hélices en operación. No sólo pueden ser causa de lesiones graves en personas y objetos por sus perfiles cortantes sino que cualquier contacto con el suelo de una de ellas, en una caída, resulta en la rotura de la pala. Además la colisión de la pala con un objeto sólido produce una elevada tensión sobre el eje del rotor pudiendo llegar a doblarlo. Es por ello que se decide invertir elevada atención en el diseño de una protección adecuada para las pruebas en el laboratorio que pueda evolucionar posteriormente en una protección de las hélices del modelo final.

Manteniendo la estructura de metacrilato del helicóptero, se procede en un primer intento a fabricar una protección en forma de anillos que protejan cada una de las hélices. Para realizar la estructura que sujete dichos anillos se fresa una pieza de policloruro de vinilo (PVC) con forma de aspa, con la intención de añadir en sus extremos unas columnas por donde guiar los aros. Esta tentativa se descarta debido a que su gran área obstaculiza el flujo de aire de los rotores disminuyendo la eficiencia de estos.

El segundo intento consiste en diseñar unos brazos imprimibles en ABS con el mismo fin de sujetar las columnas que guíen los aros (figura 4.11). Este planteamiento también termina desechándose. Aunque elimina gran parte de superficie en comparación con el primer diseño, al tener que aumentar el tamaño de las piezas para conseguir la robustez deseada terminan teniendo una dimensión considerable.

Por último se realiza una estructura en forma de esfera que encierra las aspas. Además de proteger las aspas de impactos laterales hace lo propio con choques verticales. Para ello se utilizan varillas macizas de carbono de 2mm de grosor. Estas barras destacan por su elevada resistencia mecánica, baja densidad y alta flexibilidad, lo que permite que puedan doblarse hasta formar un anillo. Como queda reflejado en la imagen (figura 4.12) el helicóptero consigue quedar protegido, pero la falta de anclajes provistos en el diseño para albergar dicha protección impiden que esta sea suficientemente robusta como para aguantar golpes fuertes o caídas.

Al no encontrar la forma de incorporar adecuadamente la protección se decide crear un nuevo diseño para el helicóptero. Esta vez se tiene en cuenta la necesidad



Figura 4.11: Brazo protección ABS

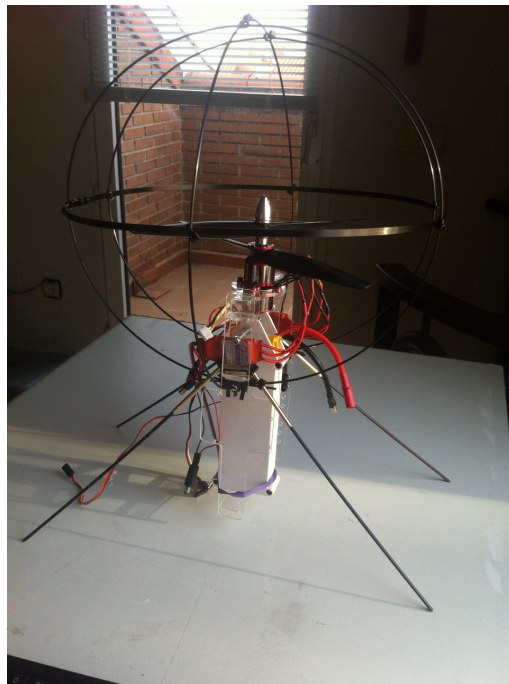


Figura 4.12: Prototipo protegido con varillas de carbono

de añadir una protección capaz de absorber caídas de la aeronave y la creación de un dispositivo de aterrizaje para que pueda realizar las maniobras de despegue y aterrizaje desde el suelo. Además deberá incluir unas guías a cada lado con la finalidad de poder restringir y controlar su vuelo mediante unos hilos introducidos en ellas.

Se barajan distintas formas de abordar el problema de la protección. La idea de crear una estructura que envuelva gran parte de la aeronave es una de ellas. Se puede construir un rombicuboctaedro con varillas de carbono en las 48 aristas y piezas impresas en sus 24 vértices. Esta figura no obstaculiza el flujo de aire de las hélices, pero se descarta por su complejidad de construcción. Se decide entonces volver al diseño de dos anillos rodeando las hélices.

En la figura (figura 4.13 (a)) se muestra un primer diseño. Este está esbozado para ser impreso, lo que le permite tener formas más complejas sin aumentar el número de piezas necesarias para ser construido, como sería necesario en un diseño con

metacrilato. Como se puede observar el diseño del chasis principal, donde se montan los componentes, se mantiene simple. La protección y el tren de aterrizaje esta formado por varillas de carbono que se sujetan mediante piezas también impresas en polímero. Al final se concluye que la simplicidad del diseño provoca una falta de robustez general. La protección no es capaz de absorber grandes energías en una colisión lo que conlleva a no ser apta.

Otro diseño más robusto es el de la figura (figura 4.13 (b)). Mantiene un concepto similar al anterior pero en este caso el esqueleto de la protección es también impreso. Se ha aumentado el grosor de todas la piezas evitando puntos débiles y zonas propensas a rotura como pueden ser los ángulos rectos. También se ha reducido el número de piezas para aminorar las conexiones frágiles entre ellas. Este diseño se refina y mejora, eliminando material para aligerar la aeronave (figura 4.13 (c)).

Cabe destacar en este diseño la carcasa de la batería. Al contrario que en los anteriores prototipos todos los componentes exceptuando los motores y el servo, que necesitan estar fijos al chasis principal, van montados sobre la parte móvil. Esto permite realizar desplazamientos del centro de gravedad con mayor eficacia al ser desplazable mayor masa de la aeronave.

Otra característica es el tren de aterrizaje. Para aumentar la capacidad de absorción de energía se emplean dos varillas de carbono introducidas a través de unas guías en forma de U. Así se evita la necesidad de cortar cuatro segmentos para cada una de las patas y se consigue aprovechar la flexibilidad y resistencia de las varillas largas.

Este último prototipo se construye pero la fabricación de las piezas en poliláctico (PLA) no genera la rigidez estructural buscada.

Finalmente se construye un modelo que aprovecha la rigidez del metacrilato con la ventaja del modelado por deposición fundida del plástico PLA (figuras 4.13 (d) y 4.14). Se incluyen en este modelo una cúpula para proteger los rotores frente a choques superiores y una base para sujetar el teléfono móvil. Este último aspecto se decide por la necesidad de poder aprovechar la cámara incorporada en el teléfono móvil para la creación de futuras aplicaciones, así como por la simplicidad para el control procedente de que el teléfono y el helicóptero compartan el sistema de referencia.

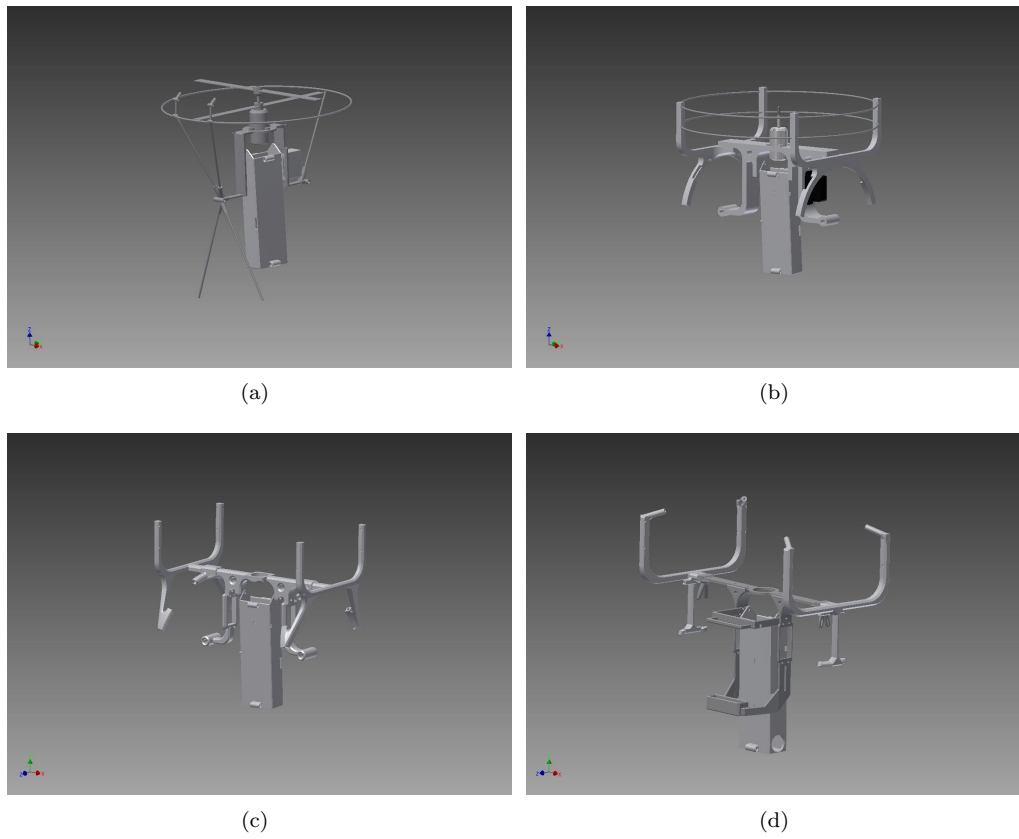


Figura 4.13: Prototipos helicóptero

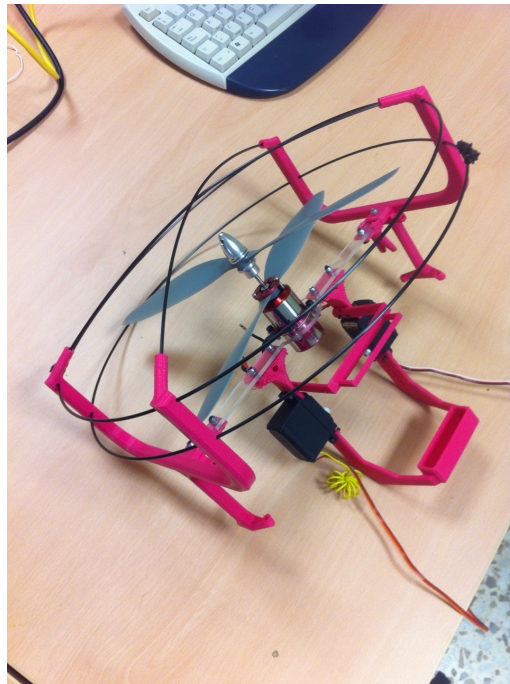


Figura 4.14: Prototipo híbrido con protección

Capítulo 5

Registro de movimiento

Para poder realizar un control apropiado sobre la aeronave, es muy importante poder registrar la variación de las variables de salida en relación al cambio en la señal de entrada de manera experimental. En el caso del helicóptero la salida principal a controlar es su posición y orientación en el espacio. En este capítulo se explica la utilización de un dispositivo láser para registrar el movimiento del helicóptero.

5.1. Láser Hokuyo

Un láser (de las siglas en inglés Light Amplification by Stimulated Emission of Radiation, amplificación de luz por emisión estimulada de radiación) es un dispositivo basado en la emisión estimulada de radiación que genera un haz de luz coherente y monocromática [3]. El Hokuyo UTM-30LX es un telémetro láser que utiliza esta tecnología para realizar mediciones de distancias de forma automática (figura 5.1). Para ello mide el tiempo que tarda un haz en rebotar sobre un objeto y calcula a partir de ese valor la distancia recorrida.

El láser Hokuyo UTM-30LX es compacto y ampliamente utilizado en aplicaciones robóticas. Entre sus principales características (tabla 5.1) destacan un campo de medida de $0,1m$ hasta $30m$ a lo largo de 270° con una velocidad de $25\frac{ms}{escaner}$.

Se adjuntan en el Anexo la hoja de características del sensor proporcionada por el fabricante (apéndice A).



Figura 5.1: Láser Hokuyo UTM-30LX

Tabla 5.1: Especificaciones láser Hokuyo UTM-30LX

Ángulo detección	270 °
Distancia detección	0,1 - 30 m
Precisión	30 - 50 mm
Resolución angular	0,25 °
Velocidad	25 ms / escáner
Interfaz	USB 2.0
Dimensiones	60 x 60 x 87 mm
Peso	370 g

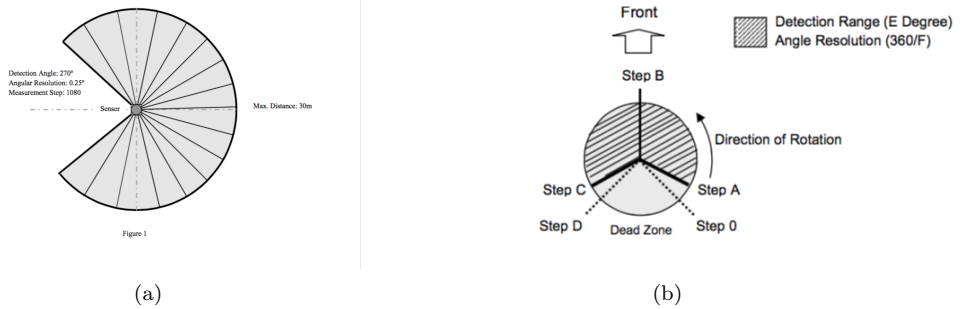


Figura 5.2: Área detección UTM-30LX

5.2. Protocolo SCIP2.0

La comunicación con el láser se realiza a través del estándar SCIP (Secure Communications Interoperability Protocol) en su variante SCIP2.0. Este protocolo fue desarrollado para permitir una interfaz segura de los sensores en aplicaciones robóticas.

Antes de realizar un programa para automatizar el proceso de medición conviene conocer algunos parámetros de la lectura del láser (figura 5.2 (a)). Cada punto de medida se conoce como paso o *step*. El láser UTM30-LX, al disponer de una resolución angular de 0,25°, realiza 1080 pasos para tomar mediciones a lo largo de su rango máximo de 270°. Si se reduce dicho rango podemos identificar el primer paso como *Step A*, el paso frontal del sensor como *Step B* y el paso final como *Step C* (figura 5.2 (b)).

Los datos recibidos en la comunicación están codificados para reducir el tiempo de transmisión entre el servidor y el sensor. Por ello necesitan ser descodificados antes de poder ser utilizados. En el protocolo SCIP2.0 se utilizan tres tipos de codificaciones de dos, tres y cuatro caracteres. Para codificar longitudes hasta 262143mm (18 bits) se utiliza la codificación de tres caracteres. En primer lugar se separa el valor de la distancia en binario en 6 bits superiores, 6 bits centrales y 6 bits inferiores. A continuación se convierten a hexadecimal y se le suma 30H al último valor, para finalmente convertir cada valor en caracteres ASCII (figura 5.3). Para descodificar dichos caracteres simplemente es necesario realizar los pasos en sentido inverso (figura 5.4).

Para comunicarse con el sensor existen trece tipos de comandos predefinidos en

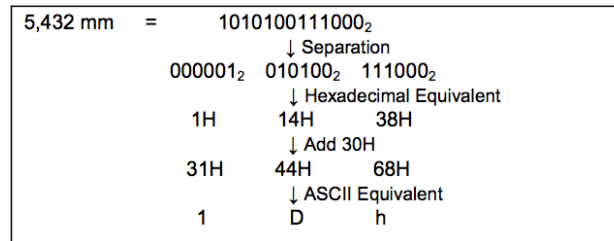


Figura 5.3: Codificación de distancias en protocolo SCIP2.0

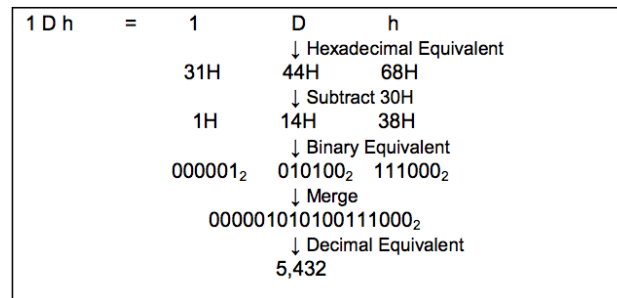


Figura 5.4: Descodificación de distancias en protocolo SCIP2.0

la especificación SCIP2.0. Todos ellos comparten el mismo formato pudiendo variar los símbolos y parámetros para realizar distintas tareas. El sensor responde a cada uno de los comandos con el formato mostrado en la figura (figura 5.5).

En la aplicación creada para registrar el movimiento del helicóptero se utiliza principalmente el comando ME. Este comando adicional para el láser UTM-30LX permite obtener valores para la distancia y la intensidad del haz recibido por el dispositivo. El comando por parte del servidor (figura 5.6 (a)) envía información del primer y último paso a escanear, el número de mediciones que se pueden fusionar en un solo paso, el intervalo para realizar las mediciones y el número de mediciones a realizar. Cuando el sensor recibe este comando responde en primer lugar con la copia del mismo comando. Esto es conocido como *Command Echo* y se utiliza para verificar posibles errores en la comunicación. A continuación añade un código de estado, útil para identificar anomalías. Y por último envía los datos de las mediciones, distancia y tiempo, codificados en tres y cuatro caracteres respectivamente y divididos en bloques de 64 bytes (figura 5.6 (b)).

Es muy importante conocer tanto el número de bytes que se reciben del sen-

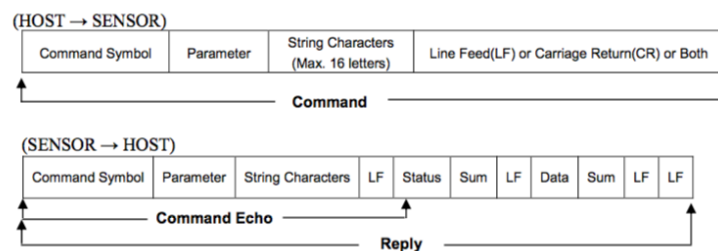


Figura 5.5: Comunicación entre servidor y sensor

(HOST→SENSOR)

'M'(4dh)	'E'(45h)		
Start Step (4byte)	End Step (4byte)	Cluster Count (2byte)	
Scan Interval (1byte)	Number of Scans (2byte)	String Characters	LF

'M'	'E'	Start Step (4byte)	End Step (4byte)	Cluster Count (2byte)	Scan Interval (1byte)
Number of Scans (2byte)	LF	String Characters	LF		
'g'	'g'	'b'	LF	Time Stamp (4 byte)	SUM (1byte)
					LF
Data Block 1 (64 bytes)	SUM (1 byte)	LF			
...			
Data Block N-1 (64 bytes)	SUM (1 byte)	LF			
Data Block N (n bytes)	SUM (1 byte)	LF	LF		

Figura 5.6: Comando ME

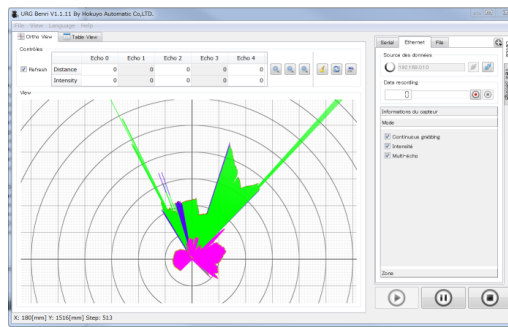


Figura 5.7: Programa URGBenri

sor como su significado para poder traducirlos a valores de distancias y tiempos aprovechables por la aplicación.

Se incluyen en el apéndice las especificaciones del estándar SCIP2.0 del fabricante Hokuyo Automatic referentes a la codificación y al formato de comunicación (apéndice A).

5.3. Captura de datos

5.3.1. UrgBenri y Excel

El fabricante Hokuyo dispone de un software gratuito para el visualizado de los datos del sensor llamado UrgBenri (figura 5.7) [14]. Este programa permite observar, grabar y reproducir de forma gráfica los datos recibidos por el láser pudiendo cambiar sus parámetros a través de una interfaz sencilla. En cambio no permite exportar los valores capturados de manera apropiada. Las grabaciones creadas pueden ser exportadas solamente en formato de archivo .ubh propio del fabricante. En cambio sí que se pueden obtener las mediciones abriendo el archivo como texto plano, sin formato, y acondicionando los valores. Para ello se utiliza la aplicación de Microsoft Excel para hojas de cálculo y su editor incorporado de VBA (Visual Basic for Applications) [20]. Este lenguaje de macros de Microsoft Visual Basic permite automatizar la tarea de ordenar los datos del archivo creado por URGBenri.

Esta macro permite finalmente obtener los valores de las medidas de manera apropiada (figura 5.8), pero la forma de conseguirlos no resulta muy oportuna. Es

	A	B	C	D	E	F	G	H	I	J
1								STEP 1	STEP 2	STEP 3
2	Tiempo	HH	MM	SS	mmm	Tiempo actual(mmm)	Tiempo (mmm)	Distancia	Distancia2	Distancia3
3	2015-03-12 19:42:20.946	19	42	20	946	70940946		0	349	6581
4	2015-03-12 19:42:20.978	19	42	20	978	70940978		32	347	6577
5	2015-03-12 19:42:21.000	19	42	21	000	70941000		54	342	6586
6	2015-03-12 19:42:21.022	19	42	21	022	70941022		76	348	6577
7	2015-03-12 19:42:21.054	19	42	21	054	70941054		108	352	6579
8	2015-03-12 19:42:21.076	19	42	21	076	70941076		130	349	6592
9	2015-03-12 19:42:21.098	19	42	21	098	70941098		152	347	6568
10	2015-03-12 19:42:21.130	19	42	21	130	70941130		184	350	6586
11	2015-03-12 19:42:21.152	19	42	21	152	70941152		206	343	6590
12	2015-03-12 19:42:21.174	19	42	21	174	70941174		228	353	6579
13	2015-03-12 19:42:21.196	19	42	21	196	70941196		250	346	6595
14	2015-03-12 19:42:21.228	19	42	21	228	70941228		282	347	6592
15	2015-03-12 19:42:21.250	19	42	21	250	70941250		304	349	6572

M	N	O	P	Q	R	S	T	U
STEP 1:	0 - 1080	Grados		appName	UrgBenri Standard			
STEP 2:	2	0.50		appVersion	1.8.7			
STEP 3:	500	125.00		model	UTM-30LX(Hokuyo Automatic Co.,Ltd.)			
	1080	270.00		frontStep	540			
				minDistance	23			
				maxDistance	60000			
				motorSpeed	2400			
				totalSteps	1440			
				captureMode	ME_Capture_mode			
				scanMsec	25			
				startStep	0			
				endStep	1080			
				grouping	1			
				serialNumber	H0904045			

Figura 5.8: Medidas mostradas en aplicación Excel

por ello que se decide crear una aplicación en Matlab capaz de automatizar todo el proceso desde la conexión al sensor hasta la exposición de los datos.

5.3.2. MATLAB

MATLAB (abreviatura de MATrix LABoratory, laboratorio de matrices) es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio, lenguaje M [17]. Entre sus principales características destacan la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. Todas ellas de gran utilidad para poder programar una aplicación como la deseada. Otra ventaja del lenguaje de alto nivel de MATLAB a la hora de programar es que no se necesitan realizar tareas administrativas de bajo nivel como declarar variables, especificar tipos de datos y asignar memoria. Además el soporte para las operaciones de vectores y matrices elimina en gran medida la necesidad de utilizar bucles. Esto permite poder escribir programas de manera más rápida y con menos líneas de código que con los lenguajes tradicionales como C o C++.

A la hora de programar en MATLAB existen dos variantes diferenciadas. Se puede crear un archivo de órdenes o *script*, que como su nombre indica es una sucesión de comandos que al llamar al archivo se ejecutan consecutivamente. O se puede crear una función que acepte parámetros de entrada y devuelva un resultado como respuesta a su llamada. Ambos tipos de archivos contienen código en lenguaje M y se guardan con extensión .m. La principal diferencia son los datos con los que trabajan. Un *script* accede a las variables del área de trabajo (*workspace*) y puede leerlas, escribir sobre ellas e incluso crear nuevas variables. En cambio una función sólo dispone de los valores pasados como parámetros y de las variables declaradas como globales. Todas las variables creadas durante su ejecución tendrán carácter de variable local, con lo que sólo perdurarán los valores devueltos por la función.

Para el desarrollo de una aplicación que automatice el proceso de obtener medidas

del sensor láser se crean cinco archivos.

El primero es un archivo de comandos llamado `SetupLidar.m`. En él se programa la conexión del láser a través del puerto serie (USB) del ordenador y se le envían al dispositivo una serie de comandos para encenderlo y conocer tanto su estado como información sobre su versión.

El segundo, `LidarScan_intensity.m`, es una función que recibe como parámetros de entrada un objeto de tipo serial y los valores del primer y último paso para las mediciones. Esta función es la encargada de enviar el comando ME descrito en apartado anterior y devolver un vector con los valores de las distancias, otro con los valores de la intensidad y el valor del tiempo. Para ello selecciona los datos útiles de la respuesta del UTM-30LX descartando los saltos de líneas y demás caracteres de comprobación. Por último envía estos datos a la función `decode_data.m`.

La función `decode_data.m` es la encargada de descodificar los valores recibidos teniendo en cuenta las codificaciones posibles de dos, tres y cuatro caracteres. Los valores de tiempo tienen una codificación de cuatro caracteres y la de la distancia e intensidad de tres caracteres.

El cuarto archivo, `DisconnectLidar.m`, vuelve a ser un *script* esta vez encargado de apagar el láser, cerrar el puerto de comunicación y destruir el objeto serie.

Se resaltan en el adjunto de este texto (apéndice A) las líneas de código más relevantes de la aplicación creada para realizar medidas con el láser Hokuyo.

Finalmente se crea un archivo que sincroniza la llamada a los archivos descritos anteriormente y encargado de representar los valores registrados en una gráfica. El origen de coordenadas representa la posición del láser, siendo x e y las coordenadas espaciales de los puntos en 2D capturados por el láser. En el eje z se representa el valor de la intensidad del haz de luz recibido en cada punto.

Al disponer del valor de la intensidad recibida se facilita la tarea de registrar el movimiento y seguimiento de puntos particulares. Esto se debe a que los puntos suelen conservar su capacidad para reflejar la radiación del haz. Para favorecer este fenómeno se puede adherir una banda reflectante sobre el objeto que se desee seguir. Así basta con observar las coordenadas con valores más elevados de z para distinguir entre el objeto y su entorno.

Se muestra a continuación una prueba realizada con la aplicación para registrar el movimiento del helicóptero coaxial. En la primera fotografía (figura 5.9) se puede observar la disposición de los elementos de este ensayo. En la izquierda de la imagen se encuentra el láser UTM-30LX montado en un lateral de la jaula. Esta jaula ha sido diseñada con la finalidad de poder realizar experimentos con gran seguridad al crear una barrera física entre la aeronave y las personas que la operan. En el centro de la imagen aparece el helicóptero. Se puede apreciar en él una banda reflectante para facilitar la interpretación de las mediciones. Por último aparece una persona encargada de restringir el movimiento de la aeronave mediante unos hilos introducidos a través de unas guías en los laterales del helicóptero.

Conociendo la colocación de las diferentes partes del ensayo, es posible analizar la gráfica creada por la aplicación de MATLAB (figura 5.10). Las dos largas líneas paralelas representan respectivamente el suelo y el techo del laboratorio. También se puede apreciar fácilmente la persona que sujeta los hilos en la nube de puntos central. Finalmente, a una distancia entre $1000mm$ y $2000mm$ del sensor, aparecen

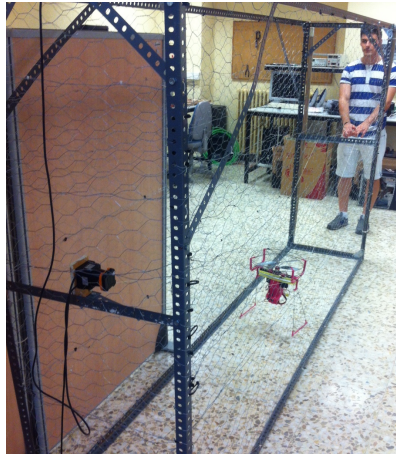


Figura 5.9: Ensayo con el helicóptero

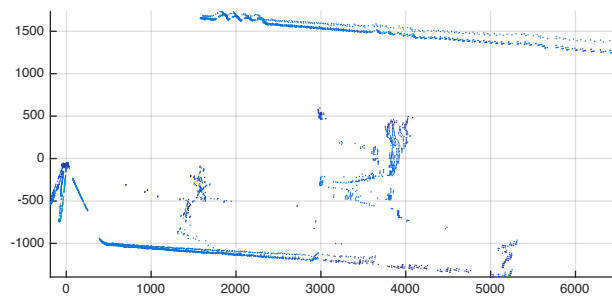


Figura 5.10: Nube de puntos en la aplicación MATLAB

los puntos referentes al helicóptero.

Visualizando la tercera dimensión del eje z concerniente a la intensidad del haz de luz recibido por el sensor láser (figura 5.11), se distinguen dos picos. El pico principal se encuentra sobre la nube de puntos del helicóptero. Esto se debe a la banda reflectante colocada sobre el helicóptero. La segunda cúspide es debida a una lámpara en el techo del laboratorio.

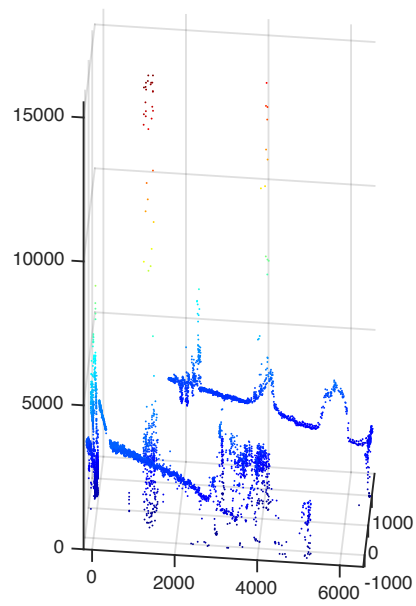


Figura 5.11: Nube de puntos en la aplicación MATLAB

Capítulo 6

Simulación del modelo dinámico del helicóptero

La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias para el correcto funcionamiento del sistema [4].

En este capítulo se describe la simulación por ordenador realizada del modelo dinámico del helicóptero con el objetivo de desarrollar un sistema de control del vuelo de la aeronave. Representando un modelo dinámico los aspectos de un sistema que cambian con el tiempo.

6.1. Simulink

Simulink es una aplicación, construida sobre MATLAB, que permite crear y simular modelos de sistemas físicos y sistemas de control mediante diagramas de bloques. El comportamiento de dichos sistemas se define mediante funciones de transferencia, operaciones matemáticas y señales predefinidas de diferentes naturalezas. Simulink dispone de utilidades que facilitan la visualización, el análisis y guardado de los resultados de la simulación y se emplea ampliamente en ingeniería de control. Igual que ocurre con MATLAB, la gestión de complementos se realiza mediante lo que se denominan *toolboxes* (cajas/paquetes de herramientas). Cada *toolbox* es un conjunto de funciones y algoritmos de cálculo especializados en un área de conocimiento: tratamiento de señales, teoría de sistemas, finanzas, etc.

Para la simulación del helicóptero se utiliza el paquete *Control System Toolbox*, específica para sistemas de control, y el paquete *Simscape* para la simulación de sistemas físicos.

6.1.1. SimMechanics

Para modelar y simular sistemas mecánicos existe una aplicación en el ambiente de Simulink llamada SimMechanics. Esta herramienta permite realizar el estudio y análisis de sistemas sin la necesidad de escribir complejas ecuaciones y modelos matemáticos. La representación de sistemas mecánicos se realiza mediante diagramas de bloques. Los bloques se unen mediante señales físicas, en vez de las señales propias de Simulink. Para comunicar un bloque de Simulink con otro de SimMechanics es

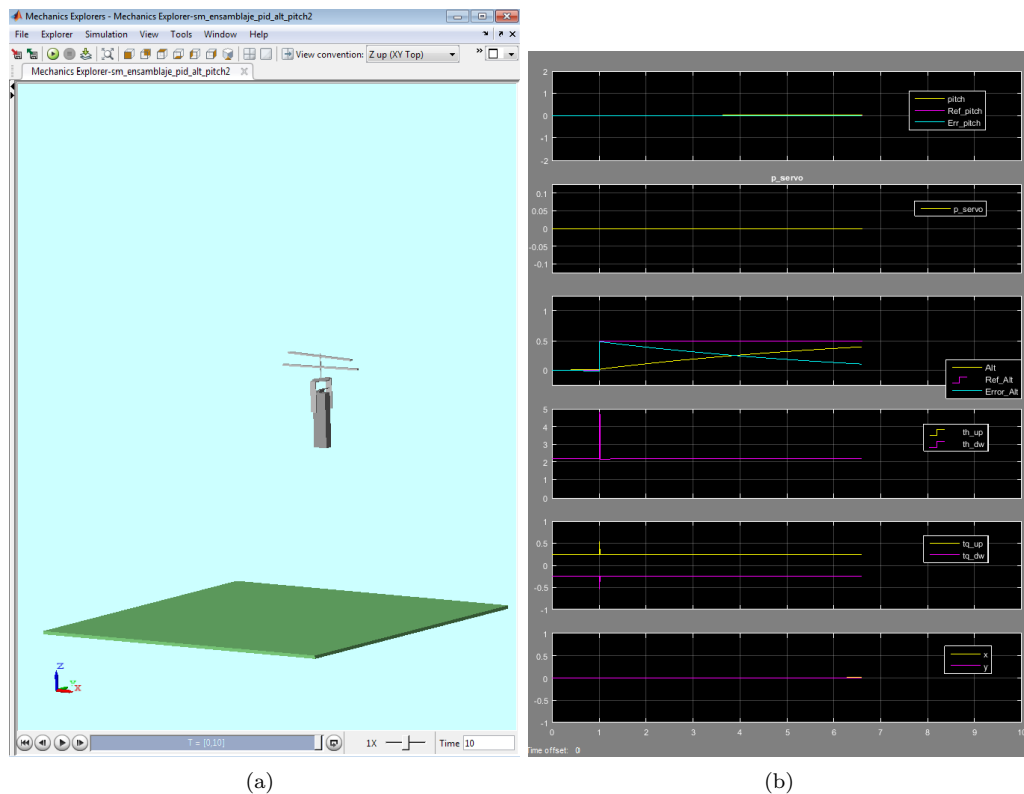


Figura 6.1: Visualización de la simulación en SimMechanics Explorer y salidas del sistema

necesario hacer uso de los bloques transductores. Estos bloques convierten la señal física en señal sin unidades utilizada por Simulink.

Un proyecto en SimMechanics se empieza definiendo las propiedades del sistema mecánico. Es necesario especificar las masas, las inercias, los grados de libertad y los sistemas de referencia de cada cuerpo. Posteriormente se colocan los actuadores y elementos de fuerza que generan el movimiento del sistema y los sensores que miden las fuerzas y movimientos de los cuerpos. Por último, se inicia la simulación donde Simulink calcula los movimientos del sistema mecánico. Los resultados de la simulación se pueden observar como animación en Mechanics Explorer y viendo las medidas obtenidas por los sensores colocados en el sistema (figura 6.1).

Como se ha comentado, se puede modelar un sistema mecánico desde SimMechanics con ayuda de las librerías de cuerpos (*bodies*) y articulaciones (*joints*). Pero también existe la posibilidad de importar directamente un modelo CAD desde una herramienta de diseño gráfico. Para ello es necesario descargar la utilidad SimMechanics Link e instalar una extensión (*plug-in*) en el software de modelado. Esta extensión permite exportar el diseño en 3D conservando la geometría, masa e inercia de cada cuerpo en formato XML. El archivo XML se importa a continuación desde SimMechanics.

Este segundo método permite aprovechar el diseño creado en Autodesk Inventor para la fabricación del helicóptero y así simular un modelo fiel al modelo real.

Una vez importado el modelo de Autodesk Inventor es necesario depurar algunos errores de la transformación del modelo CAD a un sistema mecánico. Estos fallos se encuentran principalmente en las restricciones del ensamblaje, que no se

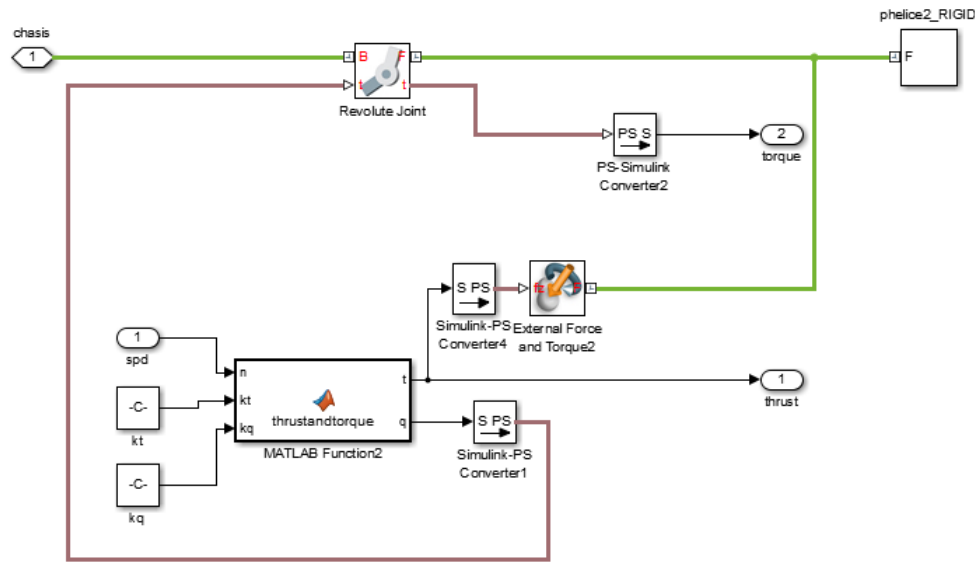


Figura 6.2: Modelo empuje hélices

corresponden con las articulaciones requeridas.

A continuación se añaden actuadores al modelo mecánico. Estos bloques se encuentran en la librería *Sensors and Actuators* y son los encargados de inicializar los movimientos de los cuerpos y las articulaciones que harán posible el funcionamiento del sistema. En la simulación del helicóptero se utilizan los actuadores de articulaciones (*Joint Actuator*). Estos permiten aplicar movimiento o par a las articulaciones de revolución encontradas tanto en el servo como en los rotores.

Para obtener resultados de la simulación se incluyen sensores que miden las velocidades de los motores, la posición del servo así como la posición y orientación del helicóptero. La señal de los sensores se envía a un bloque conocido como *scope*, que permite visualizarlas.

Un problema que surge a la hora de modelar el sistema dinámico es el de definir el empuje creado por los rotores. La complejidad de añadir un modelo aerodinámico con ayuda del paquete *Aerospace Blockset*, descarta esta solución. Se decide en su lugar calcular la fuerza de empuje resultante del giro de la hélice (sección 6.3) y aplicar dicha fuerza en el punto de anclaje entre la hélice y el eje del motor (figura 6.2).

En la figura simbólica del modelo (figura 6.3) se pueden observar los actuadores incorporados en el sistema. Aquí aparecen las velocidades en sentidos opuestos de ambas hélices y las fuerzas aplicadas en los rotores como consecuencia al giro de los álabes. También se ha indicado la fuerza del peso del helicóptero y el giro del servo capaz de desplazar el centro de gravedad de la aeronave.

A la hora de crear un diagrama de bloques complejo es conveniente dividirlo en diferentes subsistemas e ir añadiendo diferentes niveles de abstracción. Esto permite crear un sistema modular donde se facilita la modificación de las diferentes partes. También ayuda a que resulte más intuitivo navegar por el modelo pudiendo encontrar

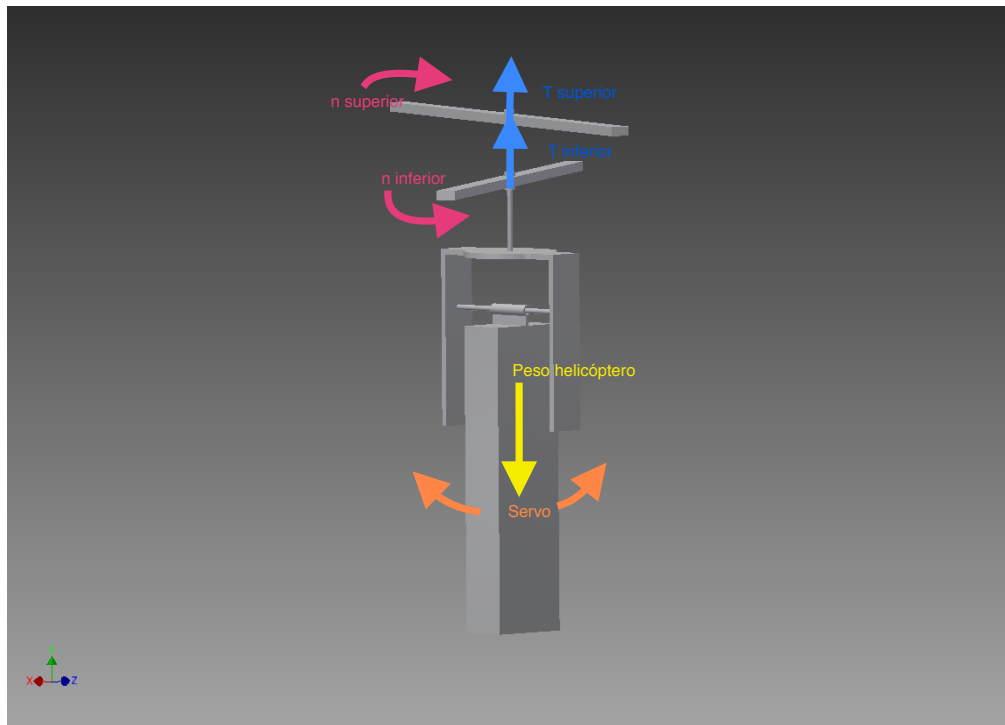


Figura 6.3: Actuadores en el modelo

rápidamente los parámetros que se desean cambiar.

En la simulación creada (figura 6.4) se puede observar como el nivel máximo de abstracción esta formado por cuatro bloques. Estos bloques representan respectivamente las entradas o consignas del sistema, los errores entre las salidas y las consignas, el controlador y por último el sistema o planta.

En el primer bloque se generan las señales de entrada deseadas para el sistema. Estas señales representan las órdenes de control mandadas por el operario que controla el helicóptero.

El segundo bloque calcula los errores existentes entre las consignas y los valores de salida medidos en la planta con ayuda de los sensores.

El bloque del controlador es el que implementa el control del sistema. Este bloque recibe los errores calculados y genera a partir de controladores PID las señales requeridas para gobernar la planta y disminuir la diferencia entre las salidas obtenidas y los valores deseados.

El cuarto bloque es el que contiene el subsistema del helicóptero. Este está formado por el modelo mecánico del helicóptero y cuenta con todos los actuadores y sensores añadidos. Las salidas de los sensores se conectan a un osciloscopio (*scope*) que permite visualizarlas.

Una vez realizado correctamente el modelo dinámico de la aeronave en el último bloque, el principal objetivo de la simulación reside en ajustar los valores del controlador presente en el tercer subsistema.

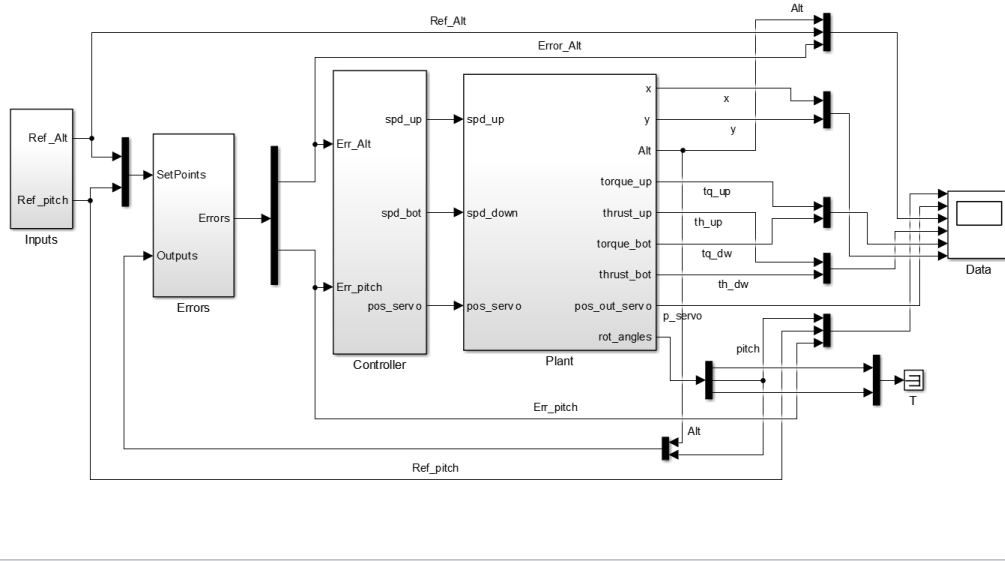


Figura 6.4: Sistema de helicóptero y control en Simulink

6.2. Regulador PID

Para realizar el control en lazo cerrado se hace uso de un regulador PID (Proporcional Integral Derivativo) por su sencillez y por ser el más utilizado en este tipo de control (figura 6.5).

Este regulador combina tres acciones mediante el siguiente algoritmo:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (F\acute{o}rmula\ te\acute{o}rica)$$

El controlador recibe la seal de error ($e(t)$) definida como la diferencia entre la consigna de la entrada ($u(t)$) y el valor de la variable de proceso a la salida de la planta ($y(t)$). Siendo la finalidad del regulador reducir dicho error con la ayuda de tres acciones. El termino proporcional depende del error presente y afecta tanto al regimen transitorio como permanente. El segundo sumando, la parte integral, es funcion de los errores acumulados pasados y tiene el efecto de reducir el error en regimen estacionario. La accion derivativa actua ante las variaciones del error pronosticando valores futuros. Por ello la accion derivativa es notoria principalmente en el estado transitorio.

La incidencia de cada una de las acciones se realiza variando las constantes proporcional (K_p), integral (K_i) y derivativa (K_d).

En el caso de anular algun termino se obtienen variantes del regulador conocidas como regulador P, PI o PD.

El proceso de ajustar las constantes en un regulador se conoce como sintona. Para ejecutarla se puede operar de dos maneras distintas. El metodo empirico permite ajustar y calcular los parametros del controlador sin conocer el modelo matematico de la planta. Como alternativa existe el metodo analitico. Este es solo aplicable si se conoce el modelo matematico de la planta a controlar y permite, a traves de

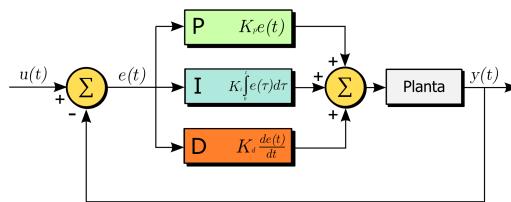


Figura 6.5: Regulador PID

técnicas de análisis temporal o frecuencial, determinar los valores óptimos para las constantes.

Ambos métodos tienen la finalidad de conseguir una respuesta específica de la planta ante una entrada dada. Esta respuesta se puede definir mediante términos como el tiempo de subida, la sobreoscilación, el tiempo de pico o el tiempo de establecimiento. Dependiendo de la naturaleza de la planta pueden existir diferentes restricciones para dichos términos debiéndose tener en cuenta durante el proceso de sintonía.

Para los modelos en Simulink se dispone de una herramienta eficaz y sencilla para realizar la sintonía del regulador PID llamada PID Tuner (figura 6.6 (a)). Esta aplicación forma parte del paquete de Control System y permite, de forma automática o interactiva, encontrar los valores para las ganancias proporcional, integral y derivativa que cumplan los requisitos de diseño.

El primer paso que realiza la aplicación es identificar la planta del proceso. Para ello simula la salida de la planta ante una entrada conocida y estima así un modelo matemático, del orden seleccionado, que describa su dinámica. A continuación permite variar la respuesta del controlador ajustando su velocidad de respuesta así como su robustez. Modificando estos dos parámetros se observa el cambio en la respuesta dibujada en una gráfica. Es necesario encontrar un equilibrio entre velocidad y robustez para conseguir las especificaciones de diseño. El programa muestra para ello en una ventana el tiempo de subida, el tiempo de establecimiento, la sobreoscilación, el tiempo de pico y los márgenes de fase y ganancia. Una vez encontradas las ganancias del PID que generan la salida deseada se puede actualizar el bloque PID del modelo.

Ahora se puede observar la respuesta del helicóptero simulando el modelo y observando su respuesta ante diversas entradas.

La gran ventaja de la simulación es la de poder ver las respuestas del helicóptero rápidamente con diferentes parámetros o configuraciones del controlador. Además cualquier error de diseño que cause inestabilidad en la aeronave y provoque que esta pierda el control no supone mayor inconveniente que el de reajustar el controlador. En cambio en un sistema real de un helicóptero, sintonías que causen inestabilidad, provocarán sin remedio un accidente.

El modelo dinámico preliminar cuenta con control de altitud ajustando la velocidad de los motores y control de inclinación variando la posición del servo que desplaza la masa principal del helicóptero (figura 6.6 (b)).

Sustituyendo los valores de la ecuación de Bernoulli (ecuación 6.2) en la ecuación del empuje (ecuación 6.1) se obtiene:

$$F = \frac{1}{2} * \rho * A * [v_s^2 - v_e^2] \quad (6.3)$$

Por otro lado, de la teoría de la conservación de la cantidad de movimiento se tiene:

$$F = [\dot{m} * v]_s - [\dot{m} * v]_0 \quad (6.4)$$

donde el sufijo s indica las condiciones a la salida, el sufijo 0 indica las condiciones del flujo libre del aire, \dot{m} es la variación de la masa a lo largo del tiempo y v la velocidad del fluido. El flujo de masa se mantiene constante en el volumen de control y se puede definir como:

$$\dot{m} = \rho * v_p * A \quad (6.5)$$

donde v_p define la velocidad del fluido. Sustituyendo esta ecuación (ecuación 6.5) en la ecuación de conservación de cantidad de movimiento (ecuación 6.4) queda:

$$F = \rho * v_p * A * [v_s - v_0] \quad (6.6)$$

Finalmente combinando ambas ecuaciones calculadas para el empuje (ecuaciones 6.3 y 6.6) se obtiene:

$$F = \frac{1}{2} * \rho * A * [v_s^2 - v_0^2] \quad (6.7)$$

siendo la velocidad del fluido:

$$v_p = \frac{1}{2} * [v_e - v_0]$$

Como se descubre en la última ecuación (ecuación 6.7), el empuje depende finalmente del cuadrado de dos velocidades difíciles de medir experimentalmente o de ser obtenidas en la simulación, como son las velocidades del aire a la entrada y salida de los rotores.

Se descarta por tanto el empleo de esta ecuación y se busca obtener otra relación entre las velocidades de las hélices y el empuje generado.

6.3.2. Análisis dimensional [26]

En este apartado se utiliza el análisis dimensional para calcular algunos coeficientes específicos de las hélices que puedan ayudar a calcular el empuje de los rotores del helicóptero y simplificar así su estudio.

Coefficiente de empuje

$$T = f(D; n; \rho; \mu; K; u_0) = c * D^a * n^b * \rho^c * \mu^d * K^e * u_0^f$$

En forma dimensional:

$$[MLT^{-2}] = [(L)^a (T^{-1})^b (ML^{-3})^c (L^2 T^{-1})^d (ML^{-1} T^{-2})^e (LT^{-1})^f]$$

Tabla 6.1: Parámetros análisis dimensional

Parámetro	Símbolo	Unidades
Diámetro hélice	D	m
Velocidad rotor	n	$\frac{rev}{s}$
Par	Q	Nm
Empuje	T	N
Densidad fluido	ρ	$\frac{kg}{m^3}$
Viscosidad fluido	μ	$\frac{kg}{m \cdot s}$
Módulo compresibilidad	K	$\frac{N}{m^2}$
Velocidad aeronave	u_0	$\frac{m}{s}$

donde M es la unidad de masa, L la unidad de longitud y T la unidad de tiempo. Para que exista consistencia dimensional se debe de cumplir:

$$\begin{aligned} (M) \quad 1 &= c + e \\ (L) \quad 1 &= a - 3c + 2d - e + f \\ (T) \quad 2 &= b + d + 2e + f \end{aligned}$$

Escogiendo las siguientes relaciones:

$$\begin{aligned} a &= 4 - 2e - 2d - f \\ b &= 2 - d - 2e - f \\ c &= 1 - e \end{aligned}$$

se obtiene:

$$T = c * D^{4-2e-2d-f} * n^{2-d-2e-f} * \rho^{1-e} * \mu^d * K^e * u_0^f$$

pudiéndose escribir como:

$$T = c * \rho * n^2 * D^4 * F \left[\left(\frac{\mu}{D^2 n} \right)^d ; \left(\frac{K}{\rho D^2 n^2} \right)^e ; \left(\frac{u_0}{D n} \right)^f \right]$$

Esta última ecuación está multiplicada por un término dependiente de tres coeficientes:

- $\frac{\mu}{D^2 n}$: Dn es proporcional a la velocidad del extremo, $\frac{\mu}{distancia * velocidad} \propto \frac{1}{Re}$;
- $\frac{K}{\rho D^2 n^2}$: $\frac{K}{\rho} = a^2$, donde a es la velocidad de propagación del sonido, $\frac{a^2}{velocidad_{extremo}^2} \propto \frac{1}{M_{extremo}^2}$;
- $\frac{u_0}{D n}$: $\frac{u_0}{n}$ es la distancia que avanza la hélice en cada revolución.

El último término se conoce como *advance ratio* en inglés, y recibe el símbolo J .

La ecuación del empuje puede ahora definirse como:

$$T = c * \rho n^2 D^4 * F[Re, M_{extremo}, J]$$

y suele expresarse de la siguiente forma:

$$T = k_T \rho n^2 D^4 \quad (6.8)$$

siendo k_T el coeficiente de empuje, que depende del diseño de la hélice.

Coeficiente de par

Siguiendo los mismos pasos que en el apartado anterior y conociendo que el par es una fuerza multiplicada por una longitud se puede obtener la siguiente ecuación:

$$Q = k_Q \rho n^2 D^5 \quad (6.9)$$

donde k_Q es el coeficiente de par y es función del diseño de la hélice.

Ambos coeficientes, k_T y k_Q , suelen obtenerse de manera experimental, aunque también existen valores tabulados para diferentes hélices así como para diferentes velocidades de rotación y velocidades del fluido de entrada en los rotores.

Como el valor exacto tanto del empuje como del par exceden las necesidades de la simulación, se decide seleccionar un valor aproximado para cada coeficiente. Se seleccionan en este caso los valores proporcionados por el fabricante de las hélices, APC Propellers [7], para una velocidad de los rotores de $11000rpm$ y una velocidad de la aeronave de $0mph$. Se puede observar en los datos facilitados por APC Propellers (figura 6.7) como estos coeficientes se mantienen sin grandes cambios en velocidades próximas a las escogidas.

PROP RPM = 9000							
V (mph)	J (Adv Ratio)	Pe	Ct	Cp	PWR (Hp)	Torque (In-Lbf)	Thrust (Lbf)
0.0	0.00	0.0000	0.1186	0.0630	0.369	2.587	3.060
2.4	0.03	0.0531	0.1180	0.0624	0.366	2.561	3.045
4.8	0.06	0.1066	0.1172	0.0617	0.362	2.534	3.025
7.2	0.08	0.1603	0.1162	0.0610	0.358	2.505	2.998
9.6	0.11	0.2141	0.1149	0.0602	0.353	2.474	2.965
PROP RPM = 10000							
V (mph)	J (Adv Ratio)	Pe	Ct	Cp	PWR (Hp)	Torque (In-Lbf)	Thrust (Lbf)
0.0	0.00	0.0000	0.1200	0.0733	0.590	3.717	3.822
2.7	0.03	0.0467	0.1194	0.0716	0.576	3.630	3.803
5.3	0.06	0.0952	0.1186	0.0699	0.562	3.543	3.779
8.0	0.08	0.1453	0.1176	0.0681	0.548	3.451	3.747
10.6	0.11	0.1968	0.1163	0.0663	0.533	3.359	3.705
PROP RPM = 11000							
V (mph)	J (Adv Ratio)	Pe	Ct	Cp	PWR (Hp)	Torque (In-Lbf)	Thrust (Lbf)
0.0	0.00	0.0000	0.1213	0.0837	0.896	5.133	4.676
2.9	0.03	0.0418	0.1208	0.0809	0.867	4.965	4.654
5.8	0.06	0.0861	0.1200	0.0781	0.837	4.793	4.626
8.8	0.08	0.1330	0.1190	0.0753	0.806	4.618	4.588
11.7	0.11	0.1824	0.1177	0.0724	0.775	4.442	4.539

Figura 6.7: Parámetros hélices APC 10x6

Capítulo 7

Aplicación BQ ZUM

En este capítulo se describe la aplicación creada para ser ejecutada en la placa BQ Zum BT-328. Esta aplicación es la encargada de recibir los comandos del teléfono a través de tecnología Bluetooth y consecuentemente controlar los motores y servos del helicóptero. Además realiza, en su primera versión, un bucle de control sobre la velocidad en el eje z, siendo este eje perpendicular al plano de las hélices.

7.1. Protocolo de comunicación

Es importante establecer un protocolo de comunicación encargado de establecer las reglas para enviar la información a través de Bluetooth. Esta guía permite que los dispositivos sean capaces de interpretar los datos recibidos así como establece la posibilidad de descubrir si ha habido errores en la comunicación. Para ello se definen la sintaxis, semántica y sincronización de los conjuntos de datos o paquetes de la comunicación.

Empezando por la sincronización se fija una cabecera que precede el envío de cada mensaje. Como la comunicación serie Bluetooth se realiza mediante el envío de bytes se pueden enviar valores que en decimal corresponden con números entre el 0 y el 255. Se decide entonces que el byte con valor 255 ejerce de cabecera. Por lo tanto, al recibir dicho byte, se entiende como el comienzo de la recepción de un nuevo mensaje. Esta característica implica que el byte 255 debe estar exclusivamente reservado para indicar el comienzo de un mensaje, no pudiendo tomar ningún otro byte del mensaje dicho valor.

A continuación se precisa el número de datos de información que contiene cada mensaje. En la aplicación creada la placa BT-328 recibe el comando de la velocidad vertical que el operario requiere para la aeronave así como la posición de cada uno de los servos. Por ello se fija la longitud de cada mensaje en tres bytes. Cada uno de estos bytes puede tomar valores entre 0 y 250 reservando los sobrantes para la cabecera y tratamiento de errores. El primer byte codifica el valor de la velocidad de ascenso y descenso representando el byte 250 la máxima velocidad de ascenso, el byte 125 la sustentación estática y el byte 0 la máxima velocidad de descenso. Para cada servo el byte 0 implica la posición mínima de rotación, el byte 125 la posición de reposo, intermedia a ambos extremos, y el byte 250 la posición de máxima rotación.

Por último se añade un código de detección de errores conocido como verificación por redundancia cíclica (CRC por sus siglas en inglés). Este código se utiliza con frecuencia en redes digitales y dispositivos de almacenamiento para detectar cambios



Figura 7.1: Composición de un mensaje del protocolo

accidentales en los datos. El valor de verificación utilizado en este caso corresponde al resto de la división de la suma de los bytes de información con 256.

$$CRC = \left(\sum \text{bytes del mensaje} \right) \% 256$$

El empleo de este código consiste en calcular el valor CRC en el emisor y enviarlo como último byte del mensaje. Por su parte el receptor, al recibir el mensaje completo, puede volver a calcular este valor y comprobar si coincide con el CRC recibido. De esta forma si cualquiera de los bytes recibidos presenta una variación en su valor con respecto a los bytes enviados el valor de verificación lo indica debiéndose así desechar el mensaje.

Al final se concluye que cada paquete está formado por un byte de cabecera, tres bytes de mensaje y un último byte de verificación (figura 7.1).

Para programar este protocolo en software se crea un bucle que retire bytes del buffer de entrada siempre que haya disponibles. Esto es necesario debido a que si el buffer se colapsa el controlador deja de tener la información actualizada y no responde correctamente. Cuando el byte leído corresponde con la cabecera, es señal de que los próximos cuatro bytes corresponden a un nuevo mensaje. Estos bytes no tienen por que llegar de manera continua, por eso se van almacenando en un vector hasta que se recibe el byte correspondiente al CRC. Una vez recibido este último byte el siguiente paso consiste en verificar la correcta recepción del mensaje calculando el CRC con los valores del mensaje recibido y comparándolo con el último byte del mensaje. Si el código de verificación coincide el mensaje será correcto y se pueden interpretar los datos. Esta interpretación consiste en escalar los valores de los bytes recibidos a valores reales o lógicos utilizados posteriormente por el programa.

Como la orden del teléfono a los servos es directa basta con escalar los bytes recibidos a las posiciones máximas y mínimas del servo haciéndolas coincidir con el ancho de pulso que las codifique. Por ejemplo, al coincidir la posición intermedia del servomotor con su ubicación de reposo, el byte 125 se traduce como $1500\mu s$ empleados para desplazarlo a esta posición.

La velocidad en cambio se escala a valores reales en $\frac{mm}{s}$ para poder utilizarla como señal de consigna para el controlador.

También se tiene en cuenta la recepción de bytes sueltos sin la llegada previa de una cabecera para desecharlos.

Se expone a continuación un diagrama donde se muestra el flujo de ejecución de la función programada (figura 7.2).

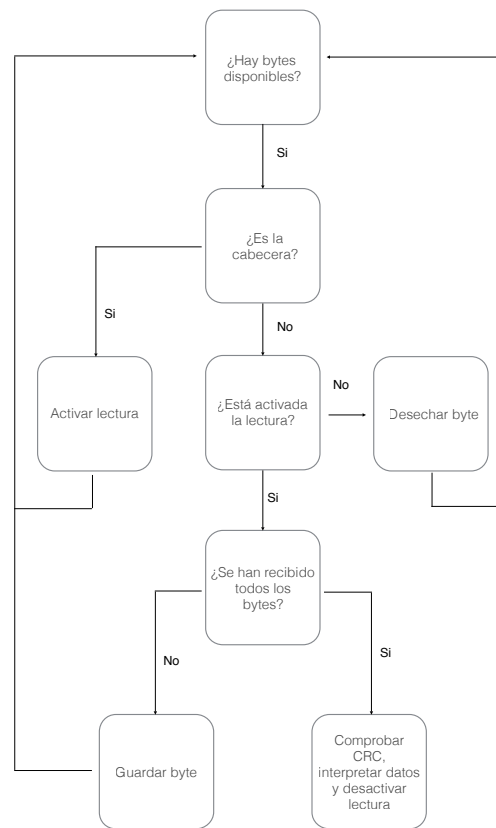


Figura 7.2: Diagrama del protocolo de comunicación

7.2. Sensor ultrasónico

Para poder realizar un control de la velocidad de ascenso y descenso del helicóptero es necesario poder medir su altura sobre el suelo. En esta aplicación se utiliza un sensor de ultrasonidos por su precisión, pequeño tamaño y precio reducido. Este detector emite ondas de ultrasonidos y mide el tiempo que tarda en regresar el eco después de rebotar sobre un objeto. El sensor escogido es el BAT Sonar Range Finder muy utilizado en aplicaciones de robótica (figura 7.3)(tabla 7.1).

Se anexan las hojas de características de este sensor en el apéndice A.

El funcionamiento de este tipo de sensores es relativamente simple. El sensor se alimenta a una tensión de $5V$ entre sus patillas de V_{cc} y GND . Para iniciar una medición hay que introducir un pulso con una duración mayor a $10\mu s$ por la entrada *Trigger* (gatillo). Como consecuencia el sensor emite ocho pulsos a una frecuencia de $40kHz$ y mide el tiempo que tarda en recibir el eco. Este tiempo lo comunica al controlador a través de la patilla *Echo* (eco) con un pulso en el que la duración equivale al tiempo medido (figura 7.4). Para conocer la distancia a la que se encuentra el objeto sobre el que ha rebotado la onda es necesario dividir el tiempo



Figura 7.3: Sensor de ultrasonidos BAT

Tabla 7.1: Especificaciones sensor ultrasonidos BAT

Rango medición	2 – 500cm
Tensión alimentación	5V
Ángulo efectivo	15°
Resolución	0,2cm

entre dos, debido a que incluye el recorrido de ida y el de vuelta, y multiplicarlo por la velocidad del sonido en el aire ($340 \frac{m}{s}$).

$$distancia\ medida[m] = \frac{duración\ del\ pulso[s] * 340 \frac{m}{s}}{2}$$

Aunque la lectura de valores es sencilla, presenta una gran dificultad a la hora de programarla en un microcontrolador sin capacidad para multiprocesamiento y en el que no se puede bloquear la ejecución debido al resto de acciones que deben ejecutarse continuamente. Como solución se emplean interrupciones para recoger los flancos de subida y bajada en la señal de eco para calcular el tiempo del pulso y varios temporizadores para sincronizar el proceso. Estos temporizadores son los encargados de medir que la señal de disparo sea mayor de $10\mu s$, que la frecuencia

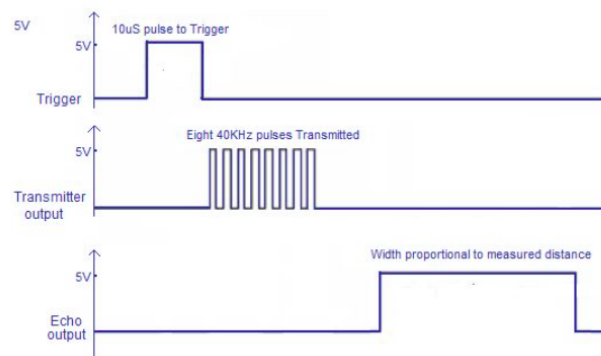


Figura 7.4: Secuencia de funcionamiento del sensor BAT

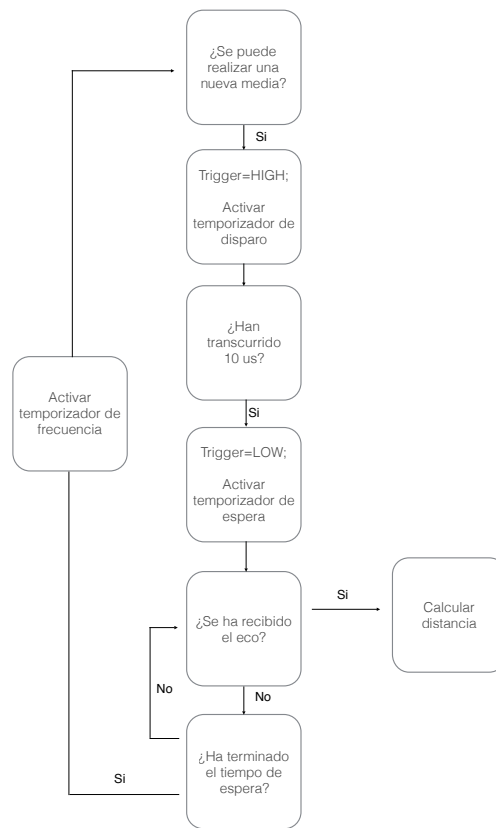


Figura 7.5: Diagrama de la medición con el sensor de ultrasonidos

de disparo no sea superior a la requerida y que se vuelva a tomar una medida si no se recibe el eco en un tiempo establecido.

En el flujograma (figura 7.5) se presenta el ciclo de la función para medir la altura con el sensor de ultrasonidos. Si durante la evolución de esta función no se cumple alguna condición, la ejecución continúa con el programa principal. Al volver a ser llamada esta función, se reanuda en la última condición no cumplida. De esta forma se impide el bloqueo de la ejecución del hilo principal evitando que se queden sin atender el resto de funciones necesarias para el vuelo del helicóptero.

Cuando se recibe un pulso del eco de una medición correcta se calcula la distancia al objeto. Estas mediciones aunque son precisas presentan bastante ruido y se recogen valores erróneos con relativa frecuencia. Para filtrar dichos errores se emplea el valor de la mediana de un conjunto de mediciones como filtro para descartar valores extremos. La mediana consiste en ordenar los valores de menor a mayor y seleccionar el valor que ocupa el lugar central.

A continuación se calcula la velocidad de la aeronave tomando las dos últimas medidas filtradas de la altura y dividiendo estas entre el intervalo de tiempo transcurrido entre ambas mediciones. En este caso el filtro empleado para la velocidad corresponde a la media móvil. Representando cada velocidad un promedio del último subconjunto de velocidades tomadas.

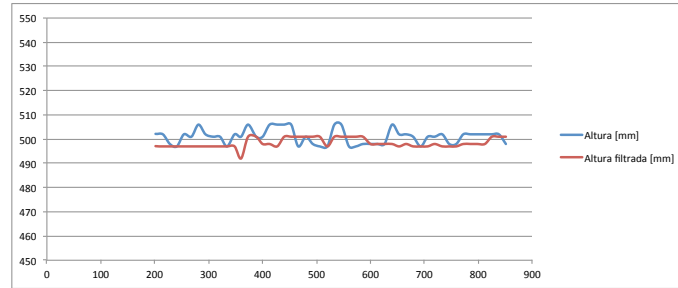


Figura 7.6: Medidas a medio metro de altura con el sensor BAT

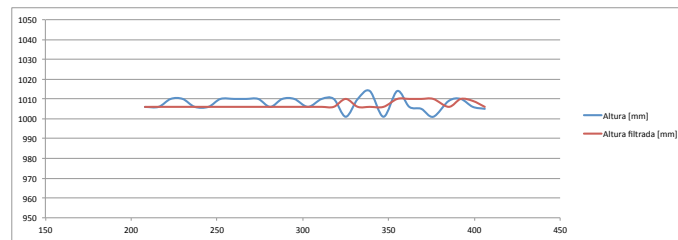


Figura 7.7: Medidas a un metro de altura con el sensor BAT

Con la aplicación terminada, se realizan medidas de alturas conocidas para determinar la precisión del sensor BAT. Estas medidas se efectúan sujetando el sensor a alturas de medio metro (figura 7.6) y un metro (figura 7.7). En las gráficas se añade también el valor de la altura filtrada. Como se puede observar, las mediciones son bastante precisas y no varían más de 10mm . Asimismo se advierte como el filtro de la mediana de once datos descarta los valores extremos.

Por último se analiza el efecto del filtrado de la velocidad. Para ello se altera constantemente la altura del sensor durante las mediciones. En la gráfica (figura 7.8) es visible el suavizado del filtrado de la media móvil de tres datos.

7.3. Control de velocidad

La siguiente función que ejecuta el programa es la de controlar la velocidad vertical con respecto a los comandos recibidos desde el teléfono móvil. Para el control en lazo cerrado se utiliza un regulador Proporcional Integral Derivativo como el descrito en la sección 6.2. Al disponer de medidas digitales, no continuas, el control se realiza con variable discreta. Esto implica que la salida del PID se calcula solamente

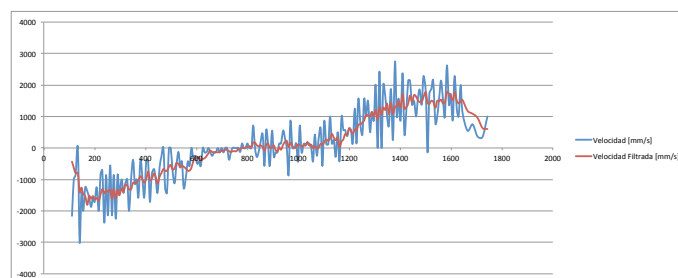


Figura 7.8: Medidas de velocidad con el sensor BAT

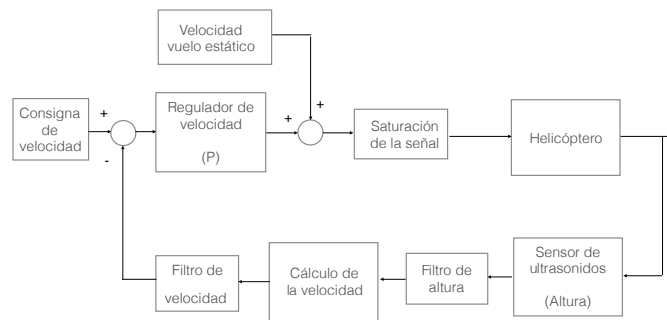


Figura 7.9: Diagrama del control de velocidad

en instantes específicos separados por un periodo de tiempo (*sample time*). Para implementar un periodo de tiempo fijo se utiliza un temporizador. Sólo cuando el temporizador lo indica se ejecuta el control.

Como primer control se utiliza exclusivamente el término proporcional, estableciendo las constantes integral y derivativa a cero. El regulador entonces calcula el error como la diferencia entre la consigna (*setpoint*) de entrada recibida desde el teléfono móvil y la variable de proceso (*process variable*) calculada a través de las mediciones del sensor de ultrasonidos. A continuación se fija la salida del regulador como el producto entre el error y la constante proporcional. En este momento puede ocurrir que en determinados instantes, donde el error es elevado, la salida del controlador tome valores no realizables. Estos valores vienen determinados como las velocidades no alcanzables por los motores. Por este motivo se satura la señal de salida del regulador permitiendo sólo el intervalo de valores posibles.

Otra característica importante es la de establecer un punto de reposo en el que la salida del regulador se aproxime a cero para condiciones estáticas en vuelo de la aeronave. Así cuando el helicóptero se encuentre en vuelo de sustentación, porque los comandos del operario así lo indiquen, el error calculado es nulo. Esto se implementa sumando la velocidad de rotación de los motores para vuelo estático en el aire a la salida del controlador.

Como consecuencia se debe volver a saturar la salida del regulador a valores comprendidos entre $1000\mu s$, correspondientes a la parada de los motores, y $2000\mu s$ establecidos para la máxima velocidad.

Una posibilidad que también se ha tenido en cuenta es que el controlador se ejecute consecutivamente sin que se hayan obtenido nuevas medidas de altura y velocidad. En este caso el controlador compara el tiempo en el que se adquirió la última medición. Si en dos ciclos del regulador el tiempo se mantiene idéntico se utiliza la velocidad cero como variable de proceso.

En la figura (figura 7.9) se representa de forma simbólica el lazo de control de la velocidad.

Los resultados del control de velocidad se muestran en la siguiente gráfica (figura 7.10). En la prueba realizada la consigna de velocidad se mantiene en $0 \frac{m}{s}$ mientras se varía la altura del sensor de ultrasonidos. Se puede observar como el error calculado es la diferencia entre la variable de proceso y cero y como la velocidad de los motores varía de forma proporcional a esta desigualdad.

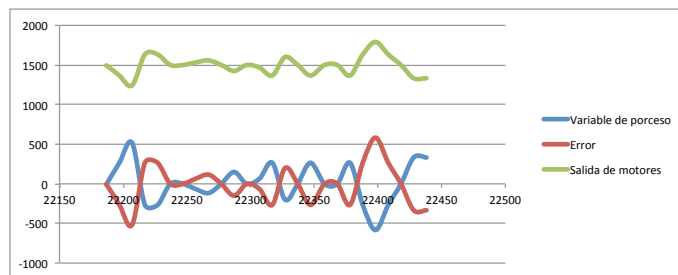


Figura 7.10: Salidas del controlador P

7.4. Bucle de la aplicación

Como se ha detallado en el apartado Programa de Arduino (sección 4.5), todos los *sketch* disponen de una función `loop()` que es ejecutada continuamente. Es en esta función por lo tanto, donde se deben realizar las llamadas al resto de funciones que constituyen el programa.

Como todas ellas necesitan ser ejecutadas con una frecuencia elevada es muy importante a la hora de programarlas que ninguna bloquee el controlador o consuma un elevado tiempo de procesamiento de este.

En primer lugar se llama a la función que procesa los datos recibidos del teléfono. Sólo conociendo las órdenes del operario se es capaz de realizar un control que responda a lo requerido. A continuación se ejecuta la función que realiza mediciones de altitud y calcula la velocidad de la aeronave para seguidamente iniciar el control. La función del controlador establece las salidas de los motores con respecto a los valores de entrada y la variable de proceso.

Por último se inician dos funciones que son las encargadas de escribir las salidas sobre los motores y los servos. La primera de ellas limita los valores que han tomado las salidas comprobando que se encuentran dentro de los límites establecidos y en caso contrario ajustándolas a los valores permitidos. La segunda función realiza el control de los actuadores proporcionándoles la señal modulada por ancho de pulso que indique la velocidad o posición que deben tomar.

Para proporcionar mayor detalle sobre la programación de las funciones que implementan el protocolo de comunicación y la toma de mediciones expuestas en este capítulo se añaden en el apéndice del presente trabajo (apéndice A) las partes más significativas del código.

Capítulo 8

Diseño del helicóptero

En este capítulo se exponen el diseño y la condición final del helicóptero.

8.1. Diseño

Después de la construcción de diversos prototipos se recogen en un diseño final lo aprendido a lo largo del presente proyecto (figura 8.1).

El último prototipo presenta, con su estructura híbrida de metacrilato y plástico PLA, el esqueleto con la mejor robustez. Se mantiene por lo tanto el soporte para el motor fresado en metacrilato. Este material posee la rigidez necesaria para servir de base para el resto del vehículo. Además absorbe las vibraciones de los motores sin provocar un exceso de fatiga en la estructura.

Las piezas de los extremos impresas para la construcción de la protección también se mantienen del último prototipo fabricado. Esta protección ha resultado ser eficaz en las pruebas realizadas. Son capaces de absorber golpes y caídas, y protegen al entorno de las hélices. Su facilidad de montaje es considerada como ventajosa. Si esta pieza se rompe en un accidente se puede volver a imprimir y colocar de nuevo en la aeronave fácilmente.

Asimismo se conserva el guiado del tren de aterrizaje. Es necesario en cambio añadir una pieza adicional para limitar la flexibilidad de las varillas de carbono. Esto se consigue fabricando dos patines que se introducen en cada par de varillas (figura 8.2). Los patines imitan a sus homólogos de los helicópteros convencionales posibilitando un aterrizaje más estable a la aeronave.

Para aumentar en mayor medida la dureza del tren de aterrizaje se añaden tirantes en cruz y en paralelo uniendo los dos patines. Los tirantes se realizan anudando hilo de nailon.

Quizás la característica principal de este nuevo diseño es su mecanismo para desplazar el centro de gravedad (figura 8.3). Por primera vez se añade un segundo grado de libertad a este movimiento. Ahora el helicóptero puede ser controlado en los tres grados de traslación en el espacio. Inclinando la masa principal, la aeronave puede realizar las maniobras de alabeo y balanceo. Para controlar la guiñada sigue siendo necesario variar la velocidad de rotación relativa de los dos rotores.

Durante la elaboración de los primeros prototipos y sus respectivas pruebas de vuelo se entiende la necesidad de poder controlar la desviación del peso de la batería en dos sentidos. Esto es imprescindible para poder contrarrestar rápidamente cualquier perturbación exterior como puede ser una ráfaga de viento lateral. Pero

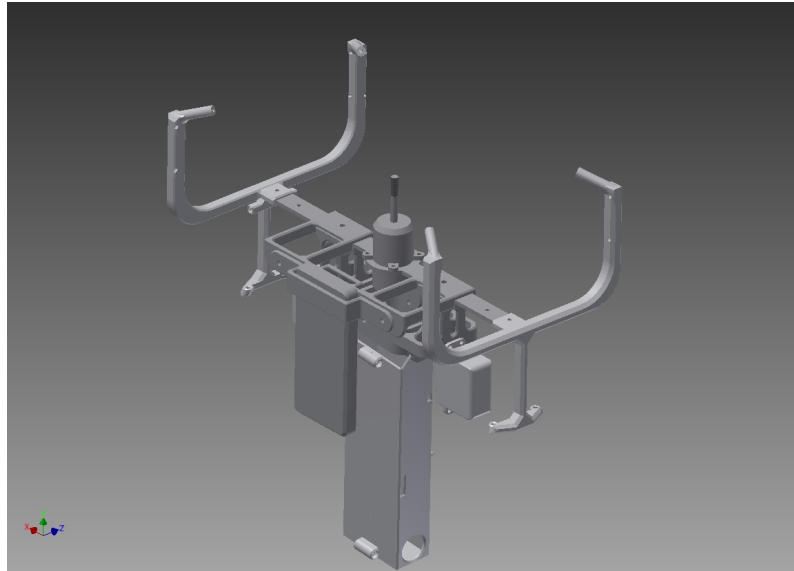


Figura 8.1: Diseño final helicóptero coaxial

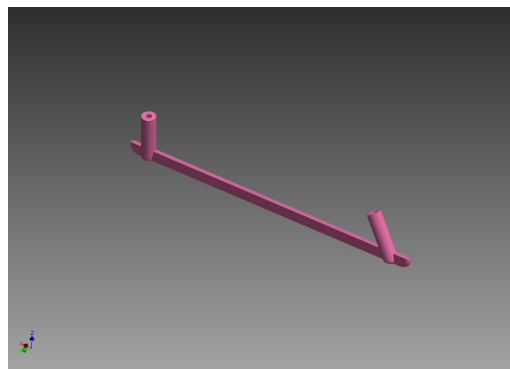


Figura 8.2: Patín del tren de aterrizaje

también es necesario para estabilizar cualquier falta de equilibrio o reparto de peso asimétrico en el helicóptero.

En el nuevo diseño se procura suprimir la existencia de holguras en los ejes de rotación de la batería. Cuanto más preciso sea el desplazamiento del centro de gravedad mejor control se puede realizar. Para su consecución se aumenta el número de paredes con sus respectivos orificios por los que introducir cada eje. Estas láminas limitan los movimientos de cada eslabón del mecanismo permitiendo bascular el peso de manera precisa.

De los diseños previos se aprende a su vez la necesidad de tener una buena sujeción donde colocar el teléfono móvil. El teléfono, al realizar los cálculos del control del helicóptero a través de sus propios sensores, exige estar fijo a la aeronave. En el último prototipo se observa como las vibraciones transmitidas al teléfono a través de su acoplamiento provocan inestabilidad en el sistema. Como solución se atornilla la nueva sujeción a la pieza central de metacrilato (figura 8.4). De esta forma se evitan las holguras existentes en el modelo anterior donde se ensambla la sujeción en los extremos de los servomotores.

Para poder alojar distintos teléfonos móviles es necesario fabricar una sencilla

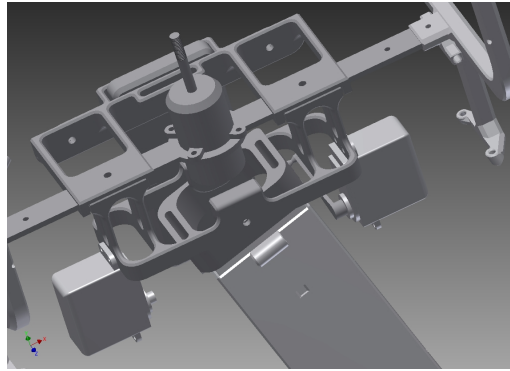


Figura 8.3: Detalle de mecanismo para desplazar el centro de gravedad

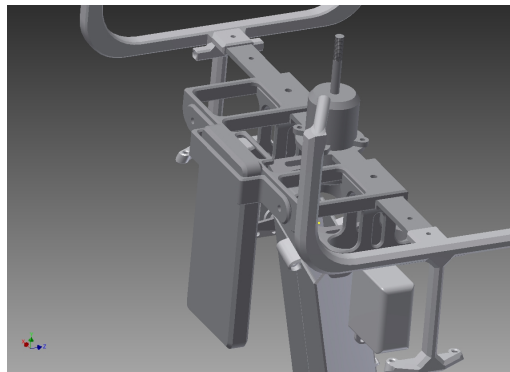


Figura 8.4: Detalle de la sujeción del teléfono móvil

pieza personalizada con el ancho del dispositivo a instalar. De esta forma se admiten diferentes teléfonos en el helicóptero sin por ello comprometer una correcta sujeción. También se ha realizado una ranura por la que introducir una cinta de velcro y así sujetar el teléfono verticalmente.

Por último, se ha desarrollado un mecanismo para desplazar lateralmente la batería. Este consiste en una articulación compuesta por una patilla con conos en sus extremos a través de los cuales se fija al brazo del servomotor y al soporte de la batería (figura 8.5). La unión entre las diferentes piezas se realiza mediante un hilo introducido en los agujeros realizados en el interior de los conos. De esta forma se consiguen obtener articulaciones que imitan a juntas esféricas con relativa facilidad. Estas juntas son necesarias debido al acoplamiento existente entre ambas rotaciones de la batería.

8.2. Control redundante de seguridad

Durante la realización de las pruebas con el helicóptero se plantea la necesidad de disponer de un sistema de seguridad para detener la aeronave en caso de pérdida de control. Todos los ensayos se ejecutan con el helicóptero volando dentro de una gran jaula y sujeto por dos guías para restringir en mayor medida su movimiento (figura 8.6). Aunque no exista riesgo para las personas, durante estas fases de desarrollo se pueden encontrar fallos en las aplicaciones programadas, no descubiertos en etapas de depuración, que conlleven algún incidente.

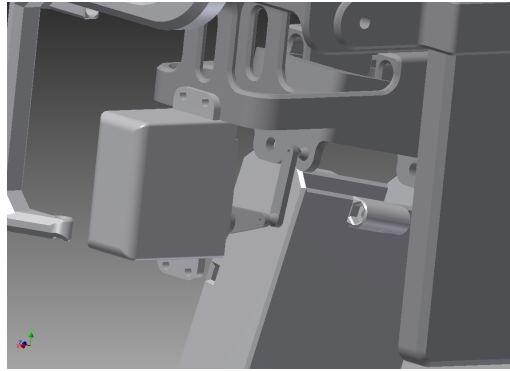


Figura 8.5: Detalle del mecanismo para desplazar la batería



Figura 8.6: Jaula para ensayos de vuelo

Por ejemplo, puede ocurrir que el control de altura pretenda volar fuera de los límites de la jaula y además se pierda la comunicación entre el teléfono móvil y el helicóptero sin poder prevenir el accidente. Por ello se decide añadir un mecanismo de seguridad que no dependa de la placa BQ ZUM.

El mecanismo diseñado cuenta además con la posibilidad de tomar el control del helicóptero con una emisora de radio. Al contrario de lo que ocurre con el vuelo de aeronaves con mayor número de rotores, como los cuadricópteros, un operario puede maniobrar este helicóptero coaxial mediante cuatro canales sin la exigencia de tener ningún tipo de controlador.

La emisora utilizada es una Turnigy TGY-i6 de seis canales. Los cuatro canales principales se manejan mediante dos palancas. Con la palanca de la izquierda se pueden controlar las velocidades absolutas y relativas de ambos motores mientras que con la palanca derecha se gobiernan los servos para el alabeo y cabeceo.

Para conmutar entre ambos controles del helicóptero se emplea un multiplexor

digital 74LS157. Este circuito integrado permite, a través de una señal de control, cambiar entre una de las dos entradas de cuatro canales siguiendo la evolución mostrada en su tabla de verdad (tabla 8.1). En esta tabla 'A' representa la primera entrada, 'B' la segunda entrada e 'Y' la salida.

Tabla 8.1: Tabla de verdad del multiplexor 74LS157

Strobe	Select	A	B	Y
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H

Se adjuntan las hojas de datos de este componente en los anexos (apéndice A).

En este momento se necesita realizar la conmutación mediante una señal externa que no dependa de la placa BT-328. Como se indica en la tabla de verdad, un valor de tensión bajo en la entrada de selección escoge la primera entrada y un valor alto de tensión la segunda entrada.

Midiendo las salidas de los pines de señal del receptor de radio se descubre que estas presentan pulsos de anchura modulada con una frecuencia de $50Hz$ cuando la emisora está encendida. Se necesita por lo tanto diseñar un circuito que produzca una señal con valor de tensión alto cuando reciba dichos pulsos a la entrada. Así, cuando se enciende la emisora, se conmuta automáticamente a sus señales de control. En cambio, mientras la emisora permanece apagada, la entrada de selección del multiplexor se encuentra con un valor bajo de tensión y se otorga el gobierno del helicóptero a la placa ZUM.

Para derivar un valor de tensión alto de una señal de pulsos con duraciones entre $1\mu s$ y $2\mu s$, se diseña un circuito tipo RC. Los pulsos recibidos se emplean para cargar el condensador y así producir un valor alto de tensión. Mientras que la resistencia en paralelo es la encargada de descargar el condensador cuando se terminen los pulsos al apagarse la emisora. Se añaden también una resistencia y un diodo en serie con la entrada para limitar la intensidad de corriente demandada del receptor de radio y para prevenir que el condensador se descargue por esta resistencia.

La resistencia que delimita la intensidad se selecciona con un valor de 500Ω para impedir intensidades mayores que $10mA$:

$$I_{max} = \frac{5V}{10mA} = 500\Omega$$

Por su parte, se escogen un condensador con capacidad de $0,1\mu F$ y una resistencia para su descarga de $1M\Omega$. El criterio para su selección es el de disponer de un tiempo de descarga de un valor alto de tensión a uno bajo en un periodo de tiempo superior a $20ms$.

$$\tau = R * C = 0,1s$$

Para construir este circuito e incorporarlo al helicóptero se realiza un diseño tipo *shield* como el realizado en el apartado 4.6. De esta forma se puede añadir la

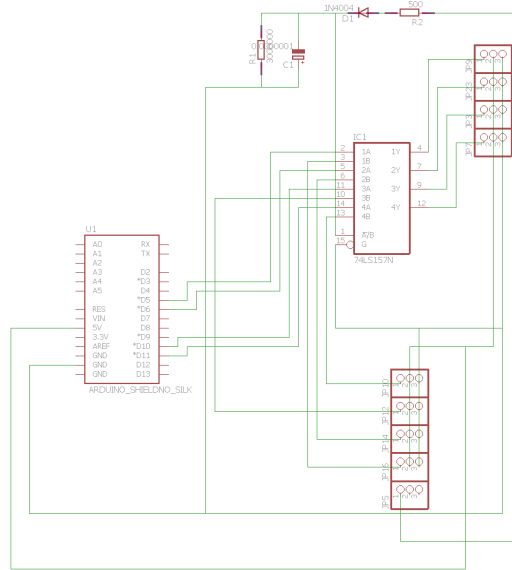


Figura 8.7: Diseño esquemático de la placa de conmutación

conmutación entre ambos controles de manera sencilla y compacta evitando tener que unir individualmente los pines de la placa BQ ZUM con el resto del circuito.

El diseño se realiza asimismo con el programa Eagle 7.3.0, añadiendo y conectando en primer lugar todos los componentes de manera esquemática (figura 8.7). Para simplificar la tarea de dimensionar la placa se utiliza la librería de la empresa Sparkfun, y en concreto la plantilla para *shields* de Arduino. A continuación se traslada el diseño esquemático al diseño de la placa. Aquí es necesario emplazar los componentes dentro de las dimensiones de la placa y trazar las pistas que los unan.

En la imagen (figura 8.8) se puede observar el diseño final. Las filas superior e inferior corresponden con los pines de entrada y salida de la placa BT-328. Los componentes electrónicos se encuentran en la parte central mientras que a su lado se localizan ocho conexiones de tipo servo. Empezando por la izquierda, las primeras cinco conexiones conciernen a las entradas del receptor de radio, siendo la quinta la conectada al circuito de conmutación. Los cuatro acoplamientos restantes coinciden con las salidas del multiplexor y es donde se conectan los motores y los servos. El resto de conexiones con la placa BQ ZUM están implícitas en el dibujo de las vías.

Capítulo 9

Gestión del proyecto

En este capítulo se describe la gestión del proyecto exponiendo la planificación seguida.

9.1. Ciclo de vida

El desarrollo del trabajo sigue una evolución similar a la representada en la redacción de este documento.

En primer lugar, al presentar los vehículos aéreos no tripulados una disciplina desconocida, se estudian sus comienzos así como la evolución hasta el estado actual. El auge presente en la actualidad de los cuadricópteros permite disponer de mucha información relativa a los componentes y tecnologías utilizadas en este tipo de aeronaves. Una vez estudiados todos los componentes se puede hacer la selección de aquellos necesarios para cumplir con las especificaciones del proyecto.

Una tarea a la que se dedica bastante tiempo es el diseño del primer banco de pruebas. Es necesario crear una estructura que permita predecir el comportamiento de la aeronave en vuelo con la particularidad de ser maniobrada mediante el desplazamiento de su centro de gravedad. La base debe permitir la observación de dicho comportamiento sin comprometer la seguridad debido a que el vehículo no debe despegar.

Una vez realizado el análisis de la dinámica se perfecciona el diseño del banco de pruebas. En este momento se cuenta con mayor experiencia para mejorar el diseño y habilidad con las herramientas para llevarlo a cabo. Por ello la construcción del segundo diseño es más rápido.

El diseño y la construcción de ambos bancos de pruebas se producen en paralelo con el aprendizaje del entorno de programación y el lenguaje de Arduino.

A continuación se detiene el progreso del helicóptero para realizar dos tareas auxiliares.

Por un lado se decide desarrollar un método para registrar el movimiento de la aeronave. Para ello se selecciona un telémetro láser. En este momento se hace necesario estudiar su protocolo de comunicación y descubrir la mejor manera para realizar las mediciones. Después de probar diferentes métodos se decide crear una aplicación específica dedicada a automatizar la comunicación con el láser y la representación de las medidas. El entorno utilizado para esta tarea es el programa MATLAB. Se hace necesario aprender el lenguaje M y programar la aplicación.

La segunda actividad auxiliar que se realiza es el desarrollo de una simulación. Esta simulación permite conocer el comportamiento del helicóptero en el aire y ayuda a definir un regulador para su control. También es necesario en este caso instruirse en el programa de Simulink y su entorno SimMechanics.

Retomando el diseño de la aeronave se realizan nuevos diseños mejorando su protección y estructura principal. El análisis de diferentes tipos de protecciones es complicado. La mayoría de vehículos aéreos no tripulados no disponen de protecciones para sus hélices. Por lo tanto se debe crear una protección personalizada para el helicóptero coaxial. Se terminan realizando varios modelos por computadora y construyendo dos de ellos.

El último modelo fabricado presenta la suficiente robustez y protección como para empezar a realizar pruebas de vuelo. A parte de disponer de protección en las hélices se introduce el helicóptero en una jaula construida expresamente para dichas pruebas. La caja es lo suficientemente grande como para que el helicóptero despegue dos metros y se desplace en una dirección. Además se colocan dos guías en el helicóptero por donde se introducen unos hilos con el fin de poder limitar una posible pérdida de control de la aeronave.

Para el control del helicóptero se programa una aplicación encargada de comunicarse con el teléfono móvil montado en el vehículo y de manejar sus motores y servos.

Como el teléfono móvil no es capaz en este momento de realizar un control de altitud se añade un sensor de ultrasonidos y un regulador PID en la placa BQ ZUM.

Por último se diseña el helicóptero definitivo con desplazamiento del centro de gravedad con dos grados de libertad.

9.2. Planificación

Para facilitar la comprensión de la evolución temporal del trabajo se expone un diagrama donde quedan reflejadas las principales tareas ejecutadas así como su duración (figura 9.1).

9.3. Presupuesto

9.3.1. Personal

El trabajo ha sido realizado por un estudiante de Grado en Ingeniería Electrónica Industrial y Automática durante 825 horas.

9.3.2. Material

Los gastos materiales del proyecto se enumeran en la siguiente tabla (tabla 9.1).

9.3.3. Resumen de costes

Todo el software utilizado en el trabajo ha sido obtenido con licencias para estudiantes y licencia Campus de la Universidad Politécnica de Madrid, por lo que no se recogen en el presupuesto.

Los costes totales corresponden exclusivamente al presupuesto del material.

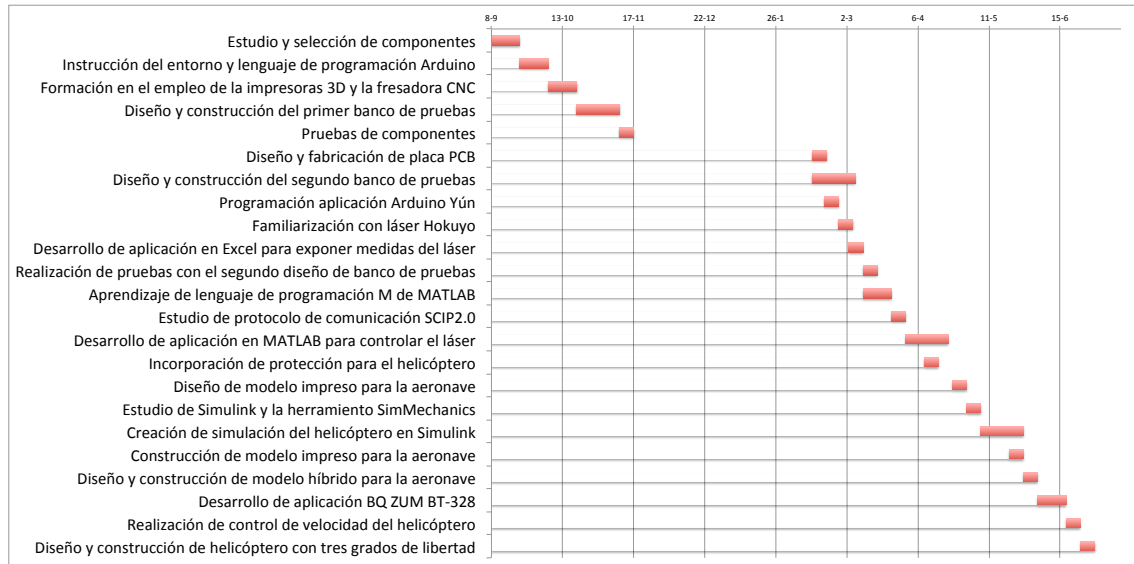


Figura 9.1: Evolución temporal del proyecto

Tabla 9.1: Presupuesto material del proyecto

Componente	Número de unidades	Precio unidad	Precio total
Motor CR28M	1	53,99 €	53,99 €
Batería Rhino 4900 mAh	1	19,06 €	19,06 €
ESC Turnigy Plush 25 A	2	12,11 €	24,22 €
Servomotor HD-3001 HB	2	12,31 €	24,62 €
BQ ZUM BT-328	1	34,90 €	34,90 €
BAT Sonar Range Finder	1	14,90 €	14,90 €
Filamento PLA BQ	1	19,90 €	19,90 €
Varillas carbono 2 mm	4	1,32 €	5,28 €
		Total:	196,87 €

Capítulo 10

Conclusiones

Se presentan a continuación las conclusiones sobre los resultados obtenidos en la elaboración del presente trabajo y se exponen posibles desarrollos futuros.

10.1. Conclusión

Una vez finalizado el proyecto se puede comprobar el cumplimiento de todos los objetivos marcados al comienzo del trabajo.

Este trabajo concluye con la construcción de un helicóptero coaxial controlado por la desviación de la posición de su centro de gravedad. El helicóptero dispone de la protección y estructura necesaria para poder ser operado tanto en zonas interiores como exteriores con completa seguridad. El diseño es robusto y soporta caídas o accidentes sin sufrir grandes daños.

Cabe destacar la exclusividad del diseño debido a la escasez de precedentes con la característica de desplazar el centro de gravedad. Como consecuencia el mecanismo para este desplazamiento es único para esta aeronave igual que la totalidad de su estructura y su protección. Por otro lado la idea de montar un teléfono en el helicóptero como plataforma para extender sus posibles aplicaciones supone también una idea innovadora no presente en los actuales vehículos aéreos no tripulados.

Por todo ello el desarrollo inicial de la aeronave ha resultado complejo. Han sido necesarios muchos diseños intermedios adaptando continuamente nuevas ideas e incluyendo mejoras con respecto a las construcciones previas. Cada nuevo diseño ha servido para entender mejor el proyecto de este helicóptero y ha servido para desarrollar una gran destreza en el diseño y desarrollo de prototipos. Esta complejidad añadida ha supuesto un desafío que aumentaba la incitación por desarrollar un buen proyecto.

Se ha dedicado gran atención en la creación de una protección eficaz tanto para el helicóptero como para poder operar con él de manera segura debido a que se entiende como parte esencial del diseño.

También se han llevado a cabo dos tareas que serán de gran utilidad en los futuros desarrollos del helicóptero.

El poder registrar los movimientos de la aeronave y disponer de estas medidas permitirá desarrollar un buen sistema de control que mejore la experiencia del vuelo con el helicóptero. Este programa, aunque se ha desarrollado con este fin, puede ser igualmente empleado en multitud de aplicaciones donde se requiera registrar movimientos con un láser que utilice el mismo protocolo SCIP2.0.

El modelo dinámico en Simulink permite simular diferentes configuraciones del regulador con gran facilidad. Pero además sirve como plantilla para importar diseños diversos de helicópteros desde herramientas de diseño asistido por computadora y visualizar su respuesta dinámica con diferentes controladores.

La aplicación de la placa BQ ZUM permite a cualquier dispositivo con tecnología Bluetooth manejar el helicóptero a través del protocolo de comunicación programado. Dispone además de un lazo de control de la velocidad vertical como función añadida. Esta aplicación es adaptable a cualquier vehículo coaxial pudiendo modificarse fácilmente distintos parámetros precisos para su control.

A parte del producto final del helicóptero se puede finalizar resaltando la gran variedad de nuevos conocimientos, de naturaleza multidisciplinar, adquiridos durante el desarrollo del trabajo sumándose a los adquiridos en la formación recibida.

10.2. Desarrollos futuros

Desde el comienzo, el presente trabajo tiene la concepción de servir de procedencia para el desarrollo de gran variedad de aplicaciones.

La incorporación de un teléfono móvil en la estructura del helicóptero tiene como objetivo principal otorgar las herramientas necesarias para la adición de diferentes habilidades al helicóptero.

Cualquier teléfono actual dispone de cámara de vídeo y fotos con la que se pueden realizar grabaciones o visualizaciones en directo desde el aire. Esto puede permitir manejar el helicóptero con imágenes en primera persona o realizar fotografías y vídeos desde el aire obteniendo perspectivas diferentes.

La gran mayoría de dispositivos móviles incorporan también sistemas de posicionamiento global (GPS) que permiten obtener la posición del terminal. Estos receptores se pueden utilizar por ejemplo en aplicaciones para vuelo autónomo pudiendo el operario definir trayectorias en un mapa que a continuación deba realizar el helicóptero.

Otra variante de desarrollo, diferente a la programación de aplicaciones para usuarios, es la de mejorar el control de la aeronave y aumentar su autonomía. Para ello se pueden utilizar mayor número de sensores presentes en el teléfono. Un buen control tiene que estimar la posición y orientación del helicóptero con gran precisión. Los datos de los diferentes sensores se pueden utilizar conjuntamente para filtrar y reducir los errores mejorando así las mediciones.

El diseño del helicóptero también puede y debe recibir constantes mejoras al tratarse de un producto nuevo y con un desarrollo breve. Cualquier vehículo de estas características debe ser retocado continuamente para suprimir defectos y añadir mejoras que se vayan descubriendo según se realicen pruebas con los prototipos.

Apéndice A

Anexos

Se presentan a continuación los anexos.

Aplicación para registro de movimiento con MATLAB

SetupLidar.m:

```
lidar=serial('COM12','baudrate',115200); % Construct a serial object.

set(lidar,'Timeout',0.1);
set(lidar,'InputBufferSize',40000);
set(lidar,'Terminator','CR');

fopen(lidar); % Connect the serial port object to the serial port.
pause(0.1);

fprintf(lidar,'BM\n'); % Switch on the laser
pause(0.1);
data = fread(lidar,lidar.BytesAvailable);
disp(char(data))
```

LidarScan_intensity.m:

```
function [ data_dist , data_int , time ] = LidarScan_intensity( lidar , start , stop )

cmd=sprintf('ME%04d%04d%02d%01d%02d\n',start,stop,0,0,nscans);
fprintf(lidar,'%s',cmd);
pause(0.01);

reply=fscanf(lidar,'%c');

for j=0:noBlocks-2
    onlyrangedata((64*j)+1:(64*j)+64)=scan(1+(66*j):(66*j)+64);
end

onlyrangedata=[onlyrangedata, scan(1+(66*(noBlocks-1)):end-3)];

encodeddist=ones(noSteps,3);
encodedint=ones(noSteps,3);

k=0;
for i=1:noSteps
    encodeddist(i,:)= [onlyrangedata((2*3*k)+1) onlyrangedata((2*3*k)+2)...
        onlyrangedata((2*3*k)+3)];
    encodedint(i,:)= [onlyrangedata((2*3*k)+4) onlyrangedata((2*3*k)+5)...
        onlyrangedata((2*3*k)+6)];
    k=k+1;
end
```

end

```
data_dist=ones(1,noSteps);
data_int=ones(1,noSteps);

for m=1:size(encodeddist,1)
    data_dist(m)=decode_data(encodeddist(m,:));
    data_int(m)=decode_data(encodedint(m,:));
end
```

decode_data.m:

```
function [ values ] = decode_data( data )

n=size(data,2);
if n==2 %2 bytes encoding
    dig1=((data(1)-'!')+33);
    dig2=((data(2)-'!')+33);
    dig1sub=dig1-48;
    dig2sub=dig2-48;
    dig1bin=dec2bin(dig1sub,6);
    dig2bin=dec2bin(dig2sub,6);
    values=bin2dec([dig1bin dig2bin]);
    return;
elseif n==3 %3 bytes encoding
    dig1=((data(1)-'!')+33);
    dig2=((data(2)-'!')+33);
    dig3=((data(3)-'!')+33);
    dig1sub=dig1-48;
    dig2sub=dig2-48;
    dig3sub=dig3-48;
    dig1bin=dec2bin(dig1sub,6);
    dig2bin=dec2bin(dig2sub,6);
    dig3bin=dec2bin(dig3sub,6);
    values=bin2dec([dig1bin dig2bin dig3bin]);
    return;
elseif n==4 %4 bytes encoding
    dig1=((data(1)-'!')+33);
    dig2=((data(2)-'!')+33);
    dig3=((data(3)-'!')+33);
    dig4=((data(4)-'!')+33);
    dig1sub=dig1-48;
    dig2sub=dig2-48;
    dig3sub=dig3-48;
    dig4sub=dig4-48;
    dig1bin=dec2bin(dig1sub,6);
    dig2bin=dec2bin(dig2sub,6);
    dig3bin=dec2bin(dig3sub,6);
    dig4bin=dec2bin(dig4sub,6);
    values=bin2dec([dig1bin dig2bin dig3bin dig4bin]);
    return;
```

Aplicación para BQ ZUM programada con Arduino IDE

Función que implementa el protocolo de comunicación:

communication.ino:

```
void readSerial() {

    // read data only when available
    while (Serial.available() > 0) {

        // read the incoming byte:
        incomingByte = Serial.read();
        LOG.INFO("\nCOM: _Byte_leido:_");
        LOG.INFO(incomingByte);
```

```

if ( incomingByte == byte(HEADER) ) {                                     // byte read is
    header
    LOG_INFO("COM: \t\t\t HEADER reconocido.");
    reading = true;
    bytesRead = 0;
}
else if ( reading == true ) {

    if (reading == true && bytesRead < (N.BYTES) ) {                      // byte read after
        header
        LOG_INFO("COM: Guardando byte...");
        LOG_INFO(bytesRead + 1);
        data[bytesRead] = incomingByte;
        bytesRead++;
    }
    else if ( reading == true && bytesRead == (N.BYTES) ) { // reading CRC
        LOG_INFO("Guardando CRC..."); // + interpreting
        data
        data[bytesRead] = incomingByte;
        int err = interpretData();
        if (err == 0) {
            LOG_INFO("\nCOM: \t\t\t FIN: Paquete recibido correctamente.");
        }
        else if (err == -1) {
            LOG_INFO("COM: ERROR RECIBIDO: CRC no coincide.");
        }
        else if (err == -2) {
            LOG_INFO("COM: ERROR RECIBIDO: valores fuera de rango.");
        }
        bytesRead = 0;
        reading = false;
    }
}

}
else { // byte received without header or out of
    range
    bytesRead = 0;
    reading = false;
    LOG_INFO("Byte recibido sin header.\n\n");
}
}
}

```

Función para realizar mediciones con el sensor de ultrasonidos y rutina de interrupción.

ultrasonic_sensor.ino:

```

int getDistance()
{
    if (frequency_timer_flag) {
        TIMER - A // FREQUENCY
        frequency_timer_previousMicros = micros();
        frequency_timer_flag = false;
    }
    frequency_timer_currentMicros = micros();
    if (frequency_timer_currentMicros - frequency_timer_previousMicros >
        TIMER.FREQUENCY) { // FREQUENCY TIMER - B

        float cm; // duration
        int mm;

        // Trigger sensor
        //=====

```

```

if (trigger_flag) {
    TRIGGER TIMER - A
    trigger_flag = false;
    trigger_previousMicros = micros();
    pulse_width = 0;

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    LOG_INFO_US("\nUS:(2) _ _ Trigger _ start");
}
else {
    trigger_currentMicros = micros();

    if (trigger_currentMicros - trigger_previousMicros > TRIGGER_INTERVAL) {
        // TRIGGER TIMER - B

        digitalWrite(trigPin, LOW); // Trigger pulse finish

        // Listen echo
        //=====

        if (timeout_timer_flag) {
            - A // TIMEOUT TIMER
            timeout_timer_flag = false;
            timeout_timer_previousMicros = micros();
        }

        timeout_timer_currentMicros = micros();

        if (pulse_width > 0) { // Echo received from
            US
            mm = microsecondsToMillimeters(pulse_width);

            LOG_INFO_US("US:(2) _ pulse_width _ != _ 0");
            LOG_INFO_US(pulse_width);
            LOG_INFO_US("US:(2) _ Distance: _");
            LOG_INFO_US(mm);
            trigger_flag = true; // Enable new trigger
            timeout_timer_flag = true; // Reset timeout timer

            return mm;
        }
        else if (timeout_timer_currentMicros - timeout_timer_previousMicros >
            TIMER_TIMEOUT) { // TIMEOUT TIMER - B
            // Timeout reached
            trigger_flag = true; // Enable new trigger
            timeout_timer_flag = true; // Reset timeout timer
            LOG_INFO_US("US:(2) _ _ _ Timer _ Reset");
        }
    }
    frequency_timer_flag = true; // Reset frequency
    timer
}
return -1;
}

void pulseChange() {
    if (digitalRead(echoPin) == HIGH) {
        prev_time = micros();
        //LOG_INFO_US("-----> Rising INTERRUPTION");
    }
    else {
        pulse_width = micros() - prev_time;
        //LOG_INFO_US("-----> Falling INTERRUPTION");
    }
}

```

}

Date: 2008.04.09

Scanning Laser Range Sensor UTM-30LX Specification

Symbol	Amendment Details			Amendment	Date
Amended by	Number				
Approved by	Checked by	Drawn by	Designed by	Title	<u>UTM-30LX</u> Specification
	MORI	KAMITANI	HINO	Drawing No	C-42-3615
					1/7



HOKUYO AUTOMATIC CO.,LTD

1. Introduction

UTM-30LX/ use laser source ($\lambda = 870\text{nm}$) to scan 270° semicircular field (Figure 1). It measures distance to objects in the range and co-ordinates of those point calculated using the step angle. Sensor's measurement data along with the angle are transmitted via communication channel.

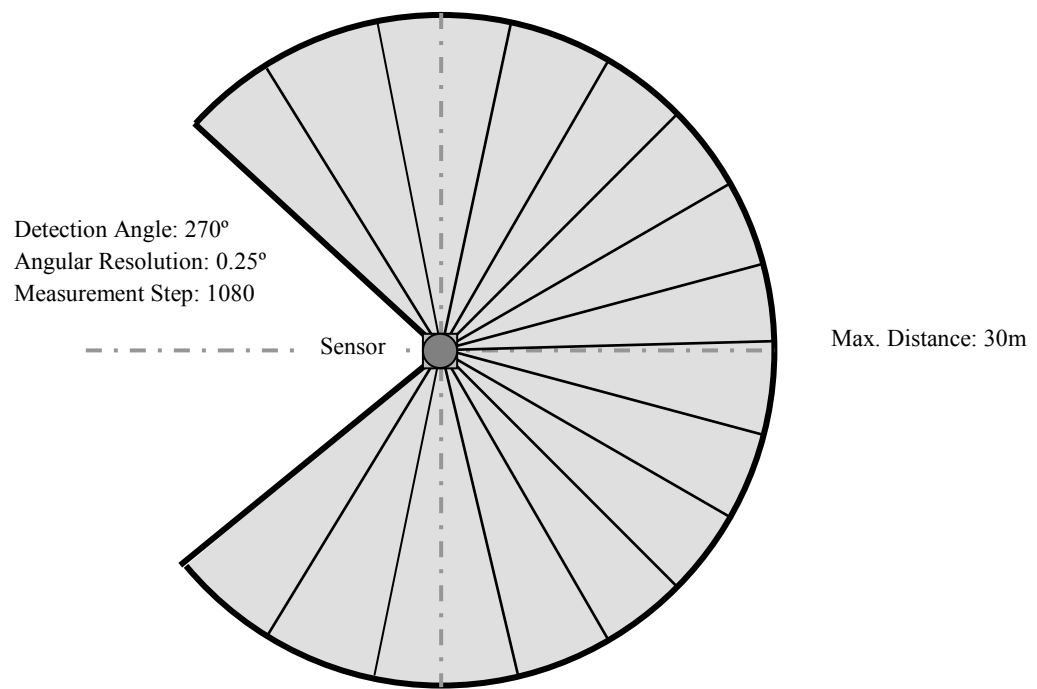


Figure 1

Title	UTM-30LX/LN Specification	Drawing No	C-42-3615	2/7
-------	---------------------------	------------	-----------	-----

2. Specifications

Product Name	Scanning Laser Range Sensor
Model	UTM-30LX
Light Source	Laser Semiconductor $\lambda = 870\text{nm}$, Laser Class 1
Supply Voltage	DC 12V $\pm 10\%$
Supply Current	Max: 1A, Normal : 0.7A
Power Consumption	Less than 8W
Detection Range and Detection Object	Guaranteed Range: 0.1 ~ 30m (White Kent Sheet) Maximum Range : 0.1 ~ 60m Minimum Width detected at 10m : 130mm (Change with distance)
Accuracy	Under 3000lx : White Kent Sheet: $\pm 30\text{mm}^{*1}$ (0.1m to 10m) Under 100000lx : White Kent Sheet: $\pm 50\text{mm}^{*1}$ (0.1m to 10m) (Also refer data sheet attached with the product.)
Measurement Resolution and Repeated Accuracy	1mm Under 3000lx : $\sigma = 10\text{mm}^{*1}$ (White Kent Sheet up to 10m) Under 100000lx : $\sigma = 50\text{mm}^{*1}$ (White Kent Sheet up to 10m)
Scan Angle	270°
Angular Resolution	0.25° (360°/1440)
Scan Speed	25ms (Motor rotation speed : 2400rpm)
Interface	USB Ver2.0 Full Speed OUTPUT : Synchronous
Output	Synchronous Output 1- Point
Ambient Condition (Temperature, Humidity)	-10°C ~ +50°C Less than 85%RH (Without Dew, Frost)
Preservation Temperature	-25~75°C
Environmental Effect	Measured distance will be shorter than the actual distance under rain, snow and direct sunlight*2.
Vibration Resistance	10 ~ 55Hz Double amplitude 1.5mm in each X, Y, Z axis for 2hrs. 55 ~ 200Hz 98m/s ² sweep of 2min in each X, Y, Z axis for 1hrs.
Impact Resistance	196m/s ² In each X, Y, Z axis 10 times.
Protective Structure	Optics: IP60
Insulation Resistance	10M Ω DC500V Megger
Weight	210g (Without cable)
Case	Polycarbonate
External Dimension (W×D×H)	60mm×60mm×85mm MC-40-3127

*1 Under Standard Test Condition (Accuracy can not be guaranteed under direct sunlight.)

*2 Confirm sensor functions under operating environment. Measures such as signal processing in LX type.

3. Quality Reference Value

Operating Vibration resistance	10~150Hz 19.6m/s ² Sweep of 2min in each X,Y,Z axis for 30min
Operating Impact resistance	49m/s ² X, Y,Z axis 10 times
Angular Speed	2 π /s (1Hz)
Angular Acceleration	$\pi/2\text{rad/s}^2$
Life	5 Years (Varies with operating conditions)
Sound Level	Less than 25dB at 300 mm
Certification	FDA Approval (21 CFR part 1040.10 and 1040.11)

Title	UTM-30LX/LN Specification	Drawing No	C-42-3615	3/7
-------	---------------------------	------------	-----------	-----



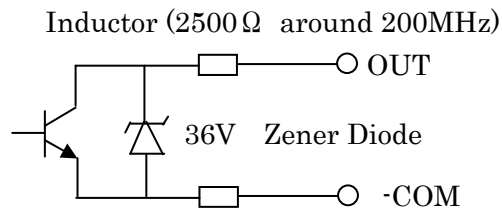
4. Interface

(1) Robot Cable 4Pin

Color	Function
Brown	+12v Power
Blue	0v Power
Green	Synchronous Output
White	COM Output (0V: Common to Power)

(2) USB Connector TYPE-A

(3) Output circuit



Note:

SG for communication and GND are connected internally (Isolated with Input -VIN).
Isolate the device from any connection that generate electric noise.
This sensor is compatible with SCIP2.0 protocol standard.

5. Control Signal

(1) Synchronous Output

Output is one pulse for approximately 1msec after every scan (Figure 2).

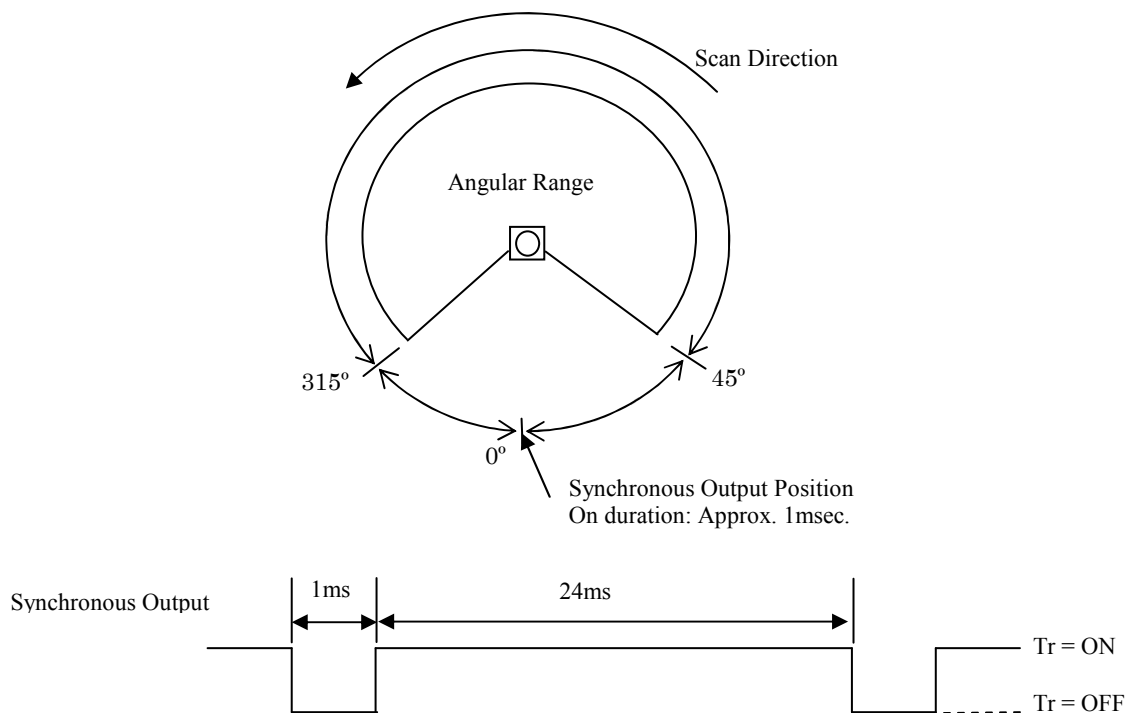


Figure 2

Title	UTM-30LX/LN Specification	Drawing No	C-42-3615	4/7
-------	---------------------------	------------	-----------	-----

(2) Malfunction Output:

1. Laser Malfunction : When laser does not radiate or exceeds safety class 1.
2. Motor Malfunction : When rotation speed is not equal to preset value.

When these malfunctions are detected synchronous/Warning signal is turned to OFF state. Error analysis can be done via communication.

6. Additional command

ME-Command

This command is used to obtain distance and intensity of the received laser pulse from the sensor. When sensor receives this command it replies with the data measured after command is processed.

Data size is double than the normal distance command (MDMS-Command) as it contains both distance and intensity measurement. It takes 2-cycles to transmit all the data.

(HOST→SENSOR)

'M'(4dh)	'E'(45h)		
Start Step (4byte)	End Step (4byte)	Cluster Count (2byte)	
Scan Interval (1byte)	Number of Scans (2byte)	String Characters	LF

(SENSOR→HOST)

(1) When status is not 99

'M'	'E'	Start Step (4byte)	End Step (4byte)	Cluster Count (2byte)	Scan Interval (1byte)
Number of Scans (2byte)		LF	String Characters	LF	
Status (2byte)		SUM (1byte)	LF	LF	

(2) When data size is less than 64 bytes

‘M’	‘E’	Start Step (4byte)	End Step (4byte)	Cluster Count (2byte)	Scan Interval (1byte)	
Number of Scans (2byte)		LF	String Characters	LF		
‘9’	‘9’	‘b’	LF	Time Stamp (4 byte)	SUM (1byte)	LF
Data* (Less than 64 bytes)		SUM (1 byte)	LF	LF		

*Data contains distance (3 byte) and intensity value (3 byte) of each measurement step.

Title	UTM-30LX/LN Specification	Drawing No	C-42-3615	5/7
-------	---------------------------	---------------	-----------	-----

(3) When data size is more than 64bytes and ends without remaining byte

‘M’	‘E’	Start Step (4byte)		End Step (4byte)		Cluster Count (2byte)		Scan Interval (1byte)	
Number of Scans (2byte)			LF	String Characters			LF		
‘9’	‘9’	‘b’	LF	Time Stamp (4 byte)		SUM (1byte)		LF	
Data Block 1 (64 bytes)			SUM (1 byte)		LF				
...						
Data Block N (64 bytes)			SUM (1 byte)		LF	LF			





(4) When data size is more than 64 bytes and ends with n-remaining bytes

‘M’	‘E’	Start Step (4byte)		End Step (4byte)		Cluster Count (2byte)		Scan Interval (1byte)	
Number of Scans (2byte)			LF	String Characters			LF		
‘9’	‘9’	‘b’	LF	Time Stamp (4 byte)		SUM (1byte)		LF	
Data Block 1 (64 bytes)			SUM (1 byte)		LF				
...						
Data Block N-1 (64 bytes)			SUM (1 byte)		LF				
Data Block N (n bytes)			SUM (1 byte)		LF		LF		

Title	UTM-30LX/LN Specification	Drawing No	C-42-3615	6/7
-------	---------------------------	---------------	-----------	-----

DATE: 10/Oct./2006

Communication Protocol Specification For SCIP2.0 Standard

	All Page Modified			2008/03/21	Kawata	PR5446
 X 8	Web Link, LED blinking after changed to SCIP2.0, MDMS-Command explanation, Bit rate as a standard.			2,4,6,9	2007/08/03	Kawata PR5325
 X 6	Published paper link, URG_Configurer, string character explanation, encode example			2,3,4,	2006/12/08	Kawata PR5205
 X 3	Explanation on F/W update mode			3, 12	2006/11/01	Kawata PR5181
SYMBOL	AMENDED REASON			PAGES	DATE	AMENDED DWG.NO
APPROVED	CHECKED	DRAFTED	DESIGNED	TITLE	Communication Protocol Specification For SCIP2.0 Standard	
Mori	Maejima	Kawata	Kawata	DRAWING NO.		
				C-42-03320B		1/25

6. Data Encoding and Decoding

Sensor's data are encoded to reduce the transmission time between host and sensor. These data should be decoded at the host side before processing them. There are three types of encoding technique applied in the sensor depending upon the data size.

- 5.1 Two-Character Encoding
- 5.2 Three-Character Encoding
- 5.3 Four-Character Encoding

6.1 Two-Character Encoding

This encoding technique is applied to express data having maximum length of 12 bits. Encoding is done by separating data into upper and lower 6 bits and then 30H is added to convert them into ASCII characters. Figure 2 and 3 shows the Two-Character encoding and decoding example.

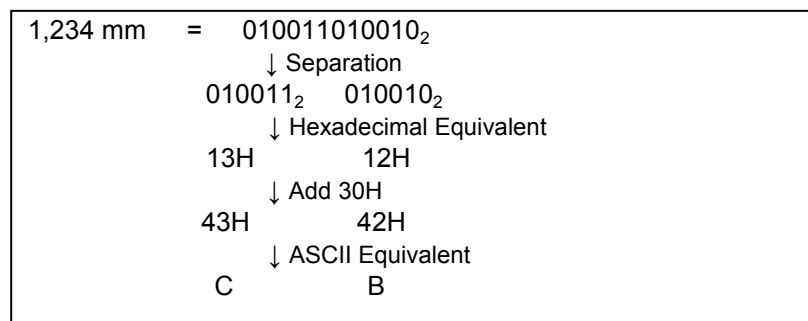


Figure 2: 2-Character Encoding Example

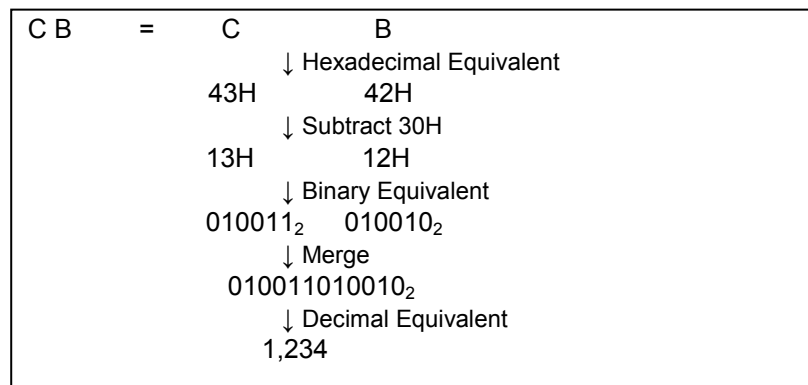


Figure 3: 2-Character Decoding Example

6.2. Three-Character Encoding

This encoding technique is applied to express data having maximum length of 18 bits. Encoding is done by separating data into upper, middle and lower 6 bits and then 30H is added to convert them into ASCII characters. Figure 4 and 5 shows the Three-Character encoding and decoding example.

5,432 mm	=	1010100111000 ₂
		↓ Separation
		000001 ₂ 010100 ₂ 111000 ₂
		↓ Hexadecimal Equivalent
		1H 14H 38H
		↓ Add 30H
		31H 44H 68H
		↓ ASCII Equivalent
		1 D h

Figure 4: 3-Character Encoding Example

1 D h	=	1	D	h
			↓ Hexadecimal Equivalent	
		31H	44H	68H
			↓ Subtract 30H	
		1H	14H	38H
			↓ Binary Equivalent	
		000001 ₂	010100 ₂	111000 ₂
			↓ Merge	
		000001010100111000 ₂		
		↓ Decimal Equivalent		
		5,432		

Figure 5: 3-Character Decoding Example

6.3. Four-Character Encoding

This encoding technique is applied to express data having maximum length of 24 bits. Encoding is done by separating data into four parts of 6 bits each and then 30H is added to convert them into ASCII characters. Figure 6 and 7 shows the Four-Character encoding and decoding example.

16,000,000 msec = 111101000010010000000000 ₂				
↓ Separation				
111101 ₂	000010 ₂	010000 ₂	000000 ₂	
↓ Hexadecimal Equivalent				
3DH	2H	10H	0H	
↓ Add 30H				
6DH	32H	40H	30H	
↓ ASCII Equivalent				
m	2	@	0	

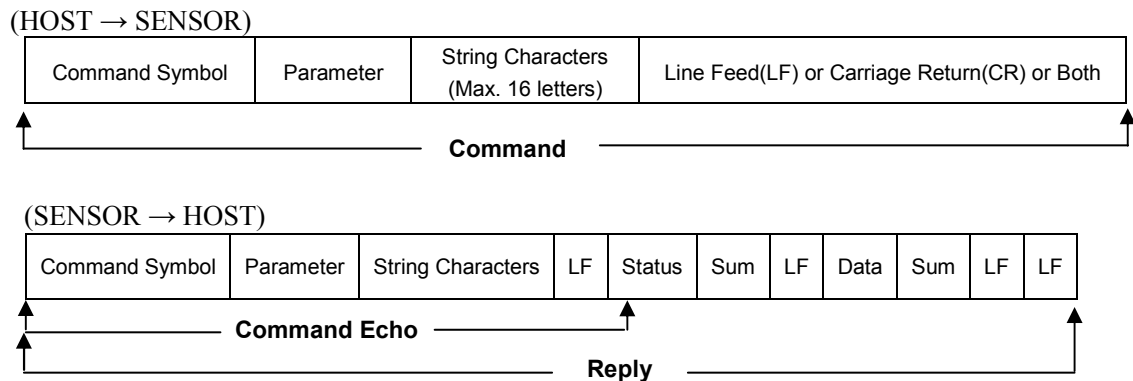
Figure 6: 3-Character Encoding Example

m2@0	=	m	2	@	0
↓ Hexadecimal Equivalent					
6DH		32H	40H	30H	
↓ Subtract 30H					
3DH		2H	10H	0H	
↓ Binary Equivalent					
111101 ₂		000010 ₂	010000 ₂	000000 ₂	
↓ Merge					
111101000010010000000000 ₂					
↓ Decimal Equivalent					
16,000,000					

Figure 7: 3-Character Decoding Example

7. Communication Format

Sensor and host exchange data using set of predefined commands. These commands have specific format known as communication format. Figure 8 and 9 shows the basic communication format between host and sensor in SCIP2.0. Communication is initiated from host by sending a **Command** that consists of Command-Symbol, Parameter, String Characters followed by line feed or carriage return or both. When sensor receive commands it sends **Reply** with the **Command Echo**, status, sum, line feed, data related to the command, sum and two continuous line feed as a termination code.



- **Command Symbol** is 2 bytes code at the beginning of every command. Each command has specific symbols for verification.
- **Parameter** is information that is required to change sensor settings or to request the additional data.
- **String Characters** are optional information in the command. They are used to verify the reply when same command is repeated more than once, such as by sending different String Characters in each command and checking the command echo. Maximum length of the String Characters is 16-letters made with combination of any English letters, numbers, blank space and symbols '.', '_', '+', '@'. They must be separated by semicolon- ';' -at the beginning to separate from parameter.
- **Line Feed (LF) or Carriage Return (CR)** is terminating code. Command can have LF or CR or both as termination code but reply will always have two continuous LF as its termination code.
- **Status** is 2 bytes data in the reply that informs normal processing if command is authenticated or errors if undefined, invalid or incomplete command is received by sensor. Status other than 00 and 99 are error codes
- **Sum** is 1 byte data used in authentication. It is calculated by adding data between two linefeeds, taking the lower 6 byte of this sum and finally adding 30H to this sum.

Example:

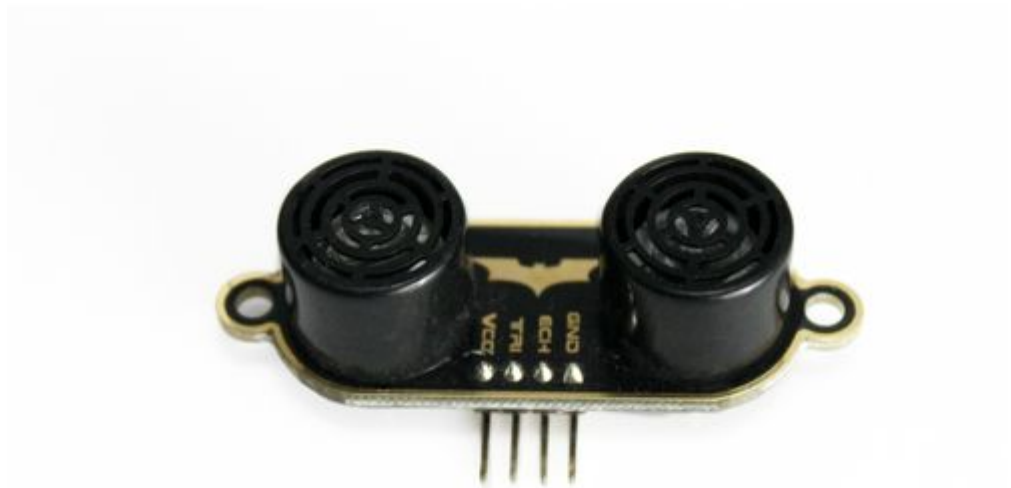
$$[\text{LF}] \text{Hokuyo} [\text{LF}] = 48\text{H} + 6\text{fH} + 6\text{bH} + 75\text{H} + 79\text{H} + 6\text{fH} = 27\text{fH} = 1001 \text{ } 111111_2$$

$$\text{Sum} = 111111_2 = 3\text{fH} + 30\text{H} = 6\text{fH} = \text{o}$$

- **Data** is main information related to the command. It is separated by LF and sum after every 64 bytes if exceeds 64 bytes.
- **IMPORTANT:** \$(24H) is a reserved letter for special mode. Do not use it in the commands.

TITLE	Communication Protocol Specification For SCIP2.0	DRAWING NO.	C-42-03320B	9/25
-------	--	----------------	--------------------	------

1. General description



BAT is a sonar range finder. It can detect objects within an area and measure the distance between BAT and objects. Ultrasonic sensors send a 40KHz high frequency sound and detect the signal reflected back from nearby objects. BAT then output a plus wich can express the range.

2. Features

- Ranging distance : 2cm - 500 cm
- Power supply: 5VDC.
- Effectual angle: 15 degree
- Resolution: 0.2cm

3. Applications

- Object detection.
- Robotic control.
- Distance measurement.

4. Electronic characteristics

PARAMETER	MIN	TYP	MAX	UNIT
Power supply voltage	4.5	5	5.5	V
Power supply current	5	100	200	mA
HIGH level input voltage	3	5	5.5	V
LOW level input voltage	-0.3	0	0.5	V

Table 1 Electronic characteristics

5. Pinning information

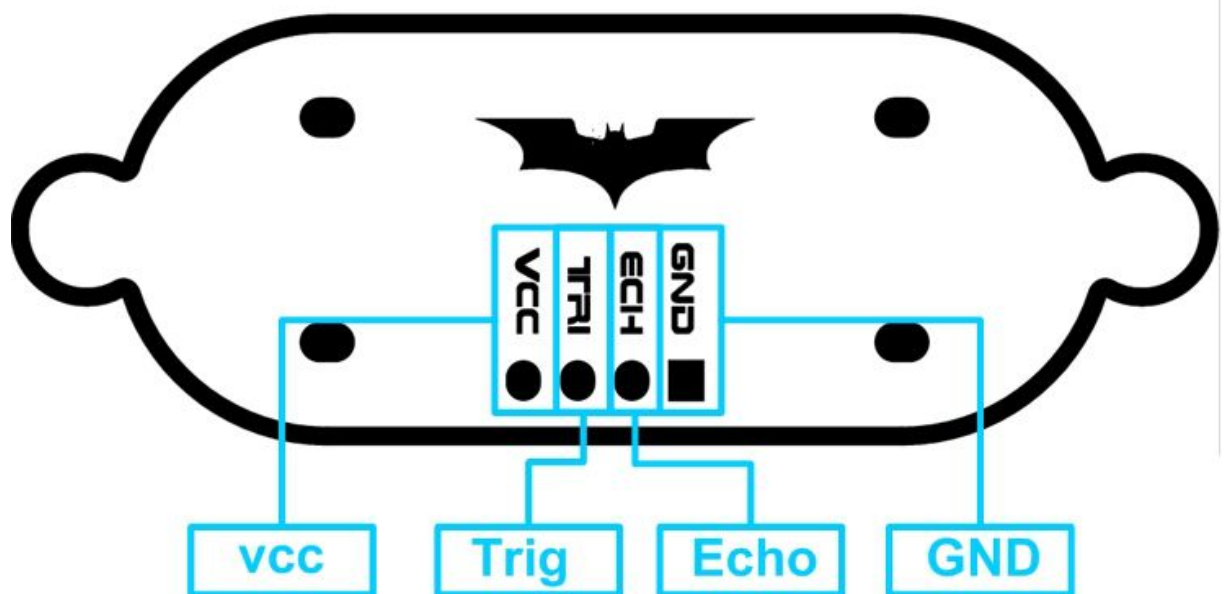


Figure 1 Top Map

TYPE	SYMBOL	DESCRIPTION
POWER	VCC	5V power supply
INPUT	Trig	A 10Us pulse is needed to trig the BAT
OUTPUT	Echo	Output a TTL high level pulse which related to distance
POWER	GND	Power Ground

Table 2 Pinning information

6. Hardware configuration

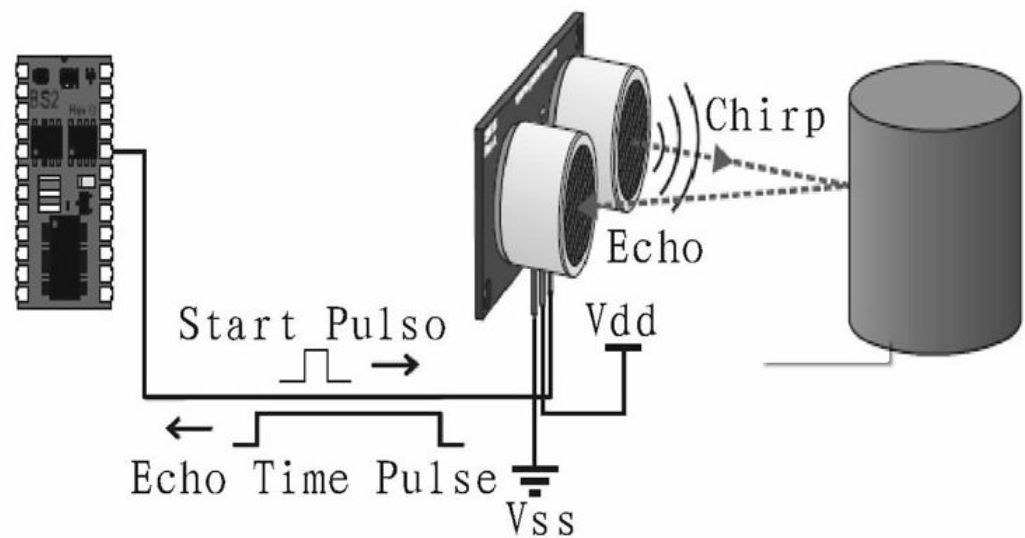


Figure 2 Schematic diagram of the testing distance

There are 4 pins out of the module : VCC , Trig, Echo, GND . So it's a very easy interface for controller to use it ranging. The all process is : pull the Trig pin to high level for more than 10us impulse , the module start ranging ; finish ranging , If you find an object in front , Echo pin will be high level , and based on the different distance, it will take the different duration of high level. So we can calculated the distance easily:

$$\text{Distance} = ((\text{Duration of high level})(\text{Sonic :340m/s}))/2$$

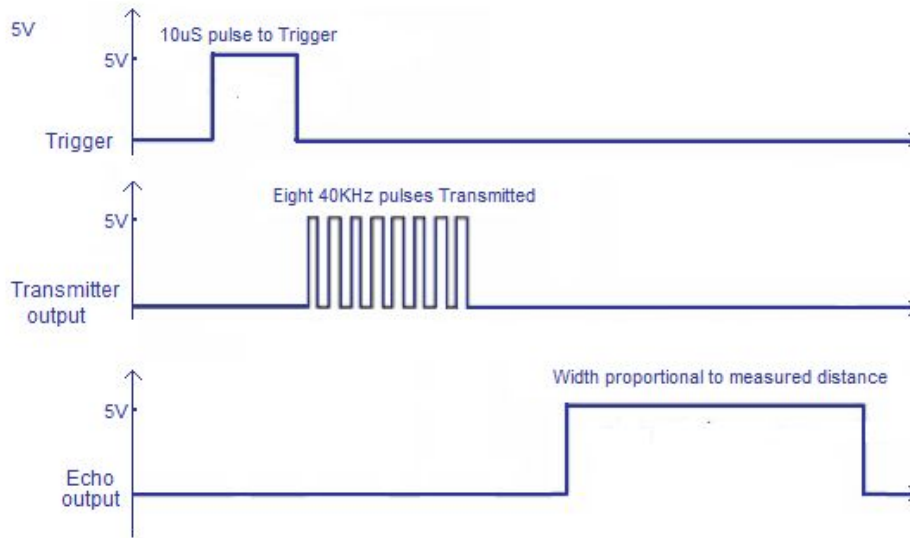


Figure 3 Signal waveform

7. Hardware connection

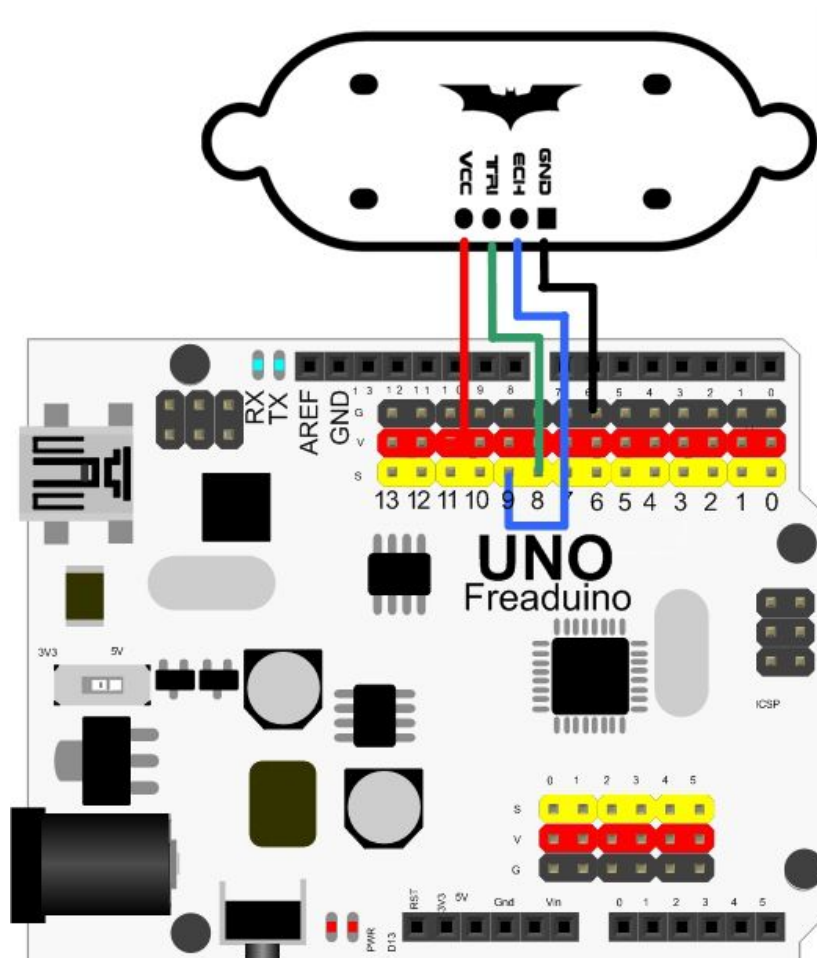


Figure 4 hardware connection



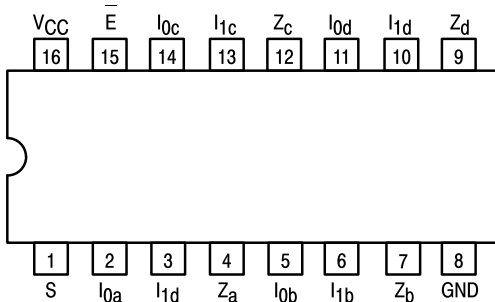
MOTOROLA

QUAD 2-INPUT MULTIPLEXER

The LSTTL/MSI SN54/74LS157 is a high speed Quad 2-Input Multiplexer. Four bits of data from two sources can be selected using the common Select and Enable inputs. The four buffered outputs present the selected data in the true (non-inverted) form. The LS157 can also be used to generate any four of the 16 different functions of two variables. The LS157 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- Schottky Process for High Speed
- Multifunction Capability
- Non-Inverting Outputs
- Input Clamp Diodes Limit High Speed Termination Effects
- Special Circuitry Ensures Glitch Free Multiplexing
- ESD > 3500 Volts

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

PIN NAMES

<u>S</u>	Common Select Input
<u>E</u>	Enable (Active LOW) Input
I0a-I0d	Data Inputs from Source 0
I1a-I1d	Data Inputs from Source 1
Za-Zd	Multiplexer Outputs (Note b)

LOADING (Note a)

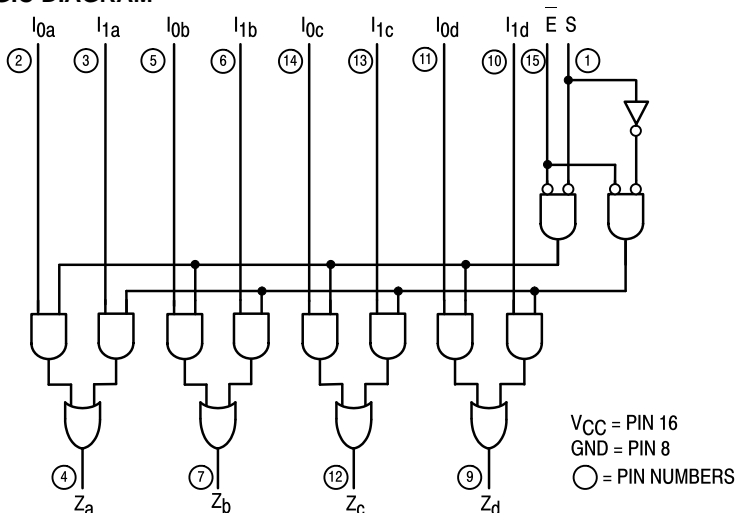
HIGH	LOW
1.0 U.L.	0.5 U.L.
1.0 U.L.	0.5 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	5 (2.5) U.L.

NOTES:

a) 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.

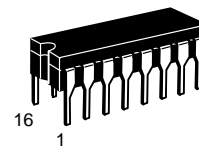
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC DIAGRAM

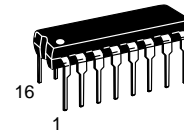


SN54/74LS157

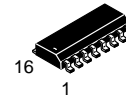
QUAD 2-INPUT MULTIPLEXER LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08

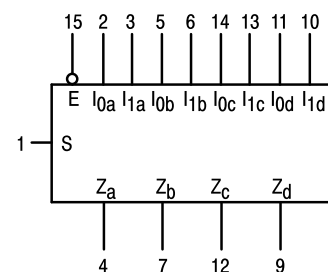


D SUFFIX
SOIC
CASE 751B-03

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

SN54/74LS157

FUNCTIONAL DESCRIPTION

The LS157 is a Quad 2-Input Multiplexer fabricated with the Schottky barrier diode process for high speed. It selects four bits of data from two sources under the control of a common Select Input (S). The Enable Input (E) is active LOW. When E is HIGH, all of the outputs (Z) are forced LOW regardless of all other inputs.

The LS157 is the logic implementation of a 4-pole, 2-position switch where the position of the switch is determined by the logic levels supplied to the Select Input. The logic equations for the outputs are:

$$\begin{aligned} Z_a &= \bar{E} \cdot (I_{1a} \cdot S + I_{0a} \cdot \bar{S}) & Z_b &= \bar{E} \cdot (I_{1b} \cdot S + I_{0b} \cdot \bar{S}) \\ Z_c &= \bar{E} \cdot (I_{1c} \cdot S + I_{0c} \cdot \bar{S}) & Z_d &= \bar{E} \cdot (I_{1d} \cdot S + I_{0d} \cdot \bar{S}) \end{aligned}$$

A common use of the LS157 is the moving of data from two groups of registers to four common output busses. The particular register from which the data comes is determined by the state of the Select Input. A less obvious use is as a function generator. The LS157 can generate any four of the 16 different functions of two variables with one variable common. This is useful for implementing highly irregular logic.

TRUTH TABLE

ENABLE	SELECT INPUT	INPUTS		OUTPUT
E	S	I ₀	I ₁	Z
H	X	X	X	L
L	H	X	L	L
L	H	X	H	H
L	L	L	X	L
L	L	H	X	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54 74			4.0 8.0	mA

SN54/74LS157

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits			Unit	Test Conditions	
			Min	Typ	Max			
V _{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V _{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V _{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V _{CC} = MIN, I _{IN} = −18 mA	
V _{OH}	Output HIGH Voltage	54	2.5	3.5		V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table	
		74	2.7	3.5		V		
V _{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I _{OL} = 4.0 mA	V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table
		74		0.35	0.5	V	I _{OL} = 8.0 mA	
I _{IH}	Input HIGH Current I _Q , I ₁ E, S				20 40	μA	V _{CC} = MAX, V _{IN} = 2.7 V	
	I _Q , I ₁ E, S				0.1 0.2	mA	V _{CC} = MAX, V _{IN} = 7.0 V	
I _{IL}	Input LOW Current I _Q , I ₁ E, S				−0.4 −0.8	mA	V _{CC} = MAX, V _{IN} = 0.4 V	
I _{OS}	Short Circuit Current (Note 1)		−20		−100	mA	V _{CC} = MAX	
I _{CC}	Power Supply Current				16	mA	V _{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS ($T_A = 25^\circ\text{C}$)

Symbol	Parameter		Limits			Unit	Test Conditions	
			Min	Typ	Max			
t_{PLH} t_{PHL}	Propagation Delay Data to Output			9.0 9.0	14 14	ns	Figure 2	$V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$
t_{PLH} t_{PHL}	Propagation Delay Enable to Output			13 14	20 21	ns	Figure 1	
t_{PLH} t_{PHL}	Propagation Delay Select to Output			15 18	23 27	ns	Figure 2	

AC WAVEFORMS

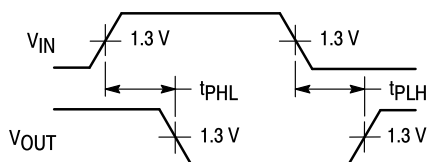


Figure 1

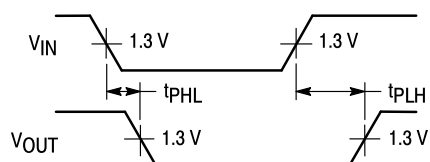


Figure 2

Bibliografía

- [1] Arduino. <https://www.arduino.cc>.
- [2] Arduino. Wikipedia, Mayo 2015. <https://es.wikipedia.org/wiki/Arduino>.
- [3] Láser. Wikipedia, Mayo 2015. <https://es.wikipedia.org/wiki/Láser>.
- [4] Simulación. Wikipedia, Mayo 2015. <https://es.wikipedia.org/wiki/Simulación>.
- [5] Vehículo aéreo no tripulado. Wikipedia, Mayo 2015. <https://es.wikipedia.org/wiki/Veh>
- [6] Aera. Flone. <http://flone.aeracoop.net>.
- [7] Apc performance data files. <https://www.apcprop.com/>.
- [8] Christian Bermes, Kevin Sartori, Dario Schafroth, Samir Bouabdallah, and Roland Siegwart. Control of a coaxial helicopter with center of gravity steering. Technical report, Autonomous Systems Lab (ASL), Swiss Federal Institute of Technology Zurich, 2008.
- [9] Samir Bouabdallah and Roland Siegwart. Design and control of an indoor coaxial helicopter. Technical report, Autonomous Systems Lab ETH Zurich, nd.
- [10] L. Chen. Modelling the lama coaxial helicopter. *Proceedings of the Australasian Conference on Robotics and Automation*, 2007.
- [11] Colin Coleman. A survey of theoretical and experimental coaxial rotor aerodynamic research. *NASA TP-3675 Technical Paper*, 1997.
- [12] Javier García de Jalón, José Ignacio Rodríguez, and Jesús Vidal. *Aprenda Matlab 7.0 como si estuviera en primero*. Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, Diciembre 2005.
- [13] Flyability. Introducing gimball, the collision-tolerant drone. <http://www.flyability.com/product/>.
- [14] Mehrez Kristou. Urgbenri information page. <https://www.hokuyo-aut.jp/02sensor/07scanner/download/data/UrgBenri.htm>, 2012.
- [15] Giuseppe Loianno, Gareth Cross, Chao Qu, Yash Mulgaonkar, Joel Hesch, and Vijay Kumar. Flying smartphones. *IEEE Robotics and Automation*, 22(2):24–32, June 2015.
- [16] Pablo Clemente Maseda. Diseño y construcción de un quadrotor de bajo coste. Technical report, Escuela Universitaria de Ingeniería Técnica Industrial, 2013.

- [17] Mathworks. Matlab, el lenguaje de cálculo técnico. <http://es.mathworks.com/products/matlab/?refresh=true>, 2015.
- [18] Mathworks. *SimMechanics User's Guide*, Marzo 2015.
- [19] Kinect for windows. <https://www.microsoft.com/en-us/kinectforwindows/>.
- [20] Microsoft. Software de hoja de cálculo. <https://products.office.com/es-es/excel>, 2015.
- [21] National Aeronautics and Space Administration. *Beginner's Guide to Propulsion*. <https://www.grc.nasa.gov/www/k-12/airplane/propth.html>.
- [22] James Paulos and Mark Yim. Flight performance of a swashplateless micro air vehicle. Technical report, University of Pennsylvania, 2015.
- [23] Eduard Petrosyan. Aerodynamic features of coaxial configuration helicopter. <http://www.kamov.ru/market/news/petr11.htm>, 2007.
- [24] A. Rudell. Advancing blade concept development test program. *AIAA/SET-P/SFTE/SAE/IEEE 1st Flight Testing Conference*, 1981.
- [25] Farid Zare Seisan. *Modeling and Control of a Co-Axial Helicopter*. PhD thesis, University of Toronto, 2012.
- [26] Prof. Z. S. Spakovszky. Thermodynamics and propulsion. Technical report, Massachusetts Institute of Technology, nd. <http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/notes.html>.
- [27] Advanced Technology and Projects group (ATAP). Project tango. <https://www.google.com/atap/project-tango/>.
- [28] Miguel A. Barcala Montejano Ángel A. Rodríguez Sevillano. *Helicópteros Teoría y Descriptiva*. Universidad Politécnica de Madrid, Escuela Universitaria de Ingeniería Técnica Aeronáutica, Abril 2008.